公用函数开发标准

2018

规范文档

TSXU

本文档所有内容均*属于*深圳市天软科技开发有限公司的商业机密，深圳市天软科技开发有限公司亦保留上述文档中阐述的包括而不限于技术、方法、思想的所有权。此文档仅仅只许可因本文档所必需传播的范围内使用（仅限于内部），本文档的获得者不得将本文档散发给不相关的其他机构和个人。

深圳天软科技 | 深圳市福田区莲花街道紫荆社区商报东路英龙商务大厦1201

# 更新日志

## 更新日志

|  |  |
| --- | --- |
| 更新日志 | 更新说明 |
| 2018-01-15 | 文档创建和发布 |
|  |  |

## 摘要

明确公用函数的开发规则，从以下6方面进行规范：

1. 函数所属目录
2. 命名
3. 参数规范
4. 函数属性设置
5. 函数体说明
6. 函数效率

# 开发标准的公用模型

每一位技术同事都担任着开发公用函数，可能是客户和业务的新需求，也可能是原有函数的升级。每个人的开发习惯都不一样，如果不标准化公用函数，那么会导致审核的同事领导花费巨额时间和精力去修改，也会导致模型使用者开发异常艰辛。

# 函数目录

函数目录应按照天软的规定进行创建，不管是提交新的公用函数，还是对原公用函数的升级，都必须注意函数的目录。下面根据新函数开发和原函数升级两种模式来说明如何创建函数目录。

## 新函数目录

新开发的函数应放在同类型函数的目录下。一般的数据计算处理类的公用函数应放在公用函数的技术指标或资产相关的目录下；数学函数应放在数学方法目录下；报表函数（应用类函数，即在生成报表或应用在某个系统显示数据的函数）应放在报表函数相对应的目录下；

注意：开发人员在开发时就需建好目录。如不清楚的，请咨询相应负责人。

下两个表列举的是现有的目录结构，如果新增的函数不属于现有的任何结构，**必须先跟负责人沟通**新增相应目录。

1、数据计算处理类的公用函数：目录为：公用函数\

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 资产相关与否 | 一级目录（按证券） | 一级命名要求 | 二级目录（按数据） | 二级命名要求 | 目录 |
| 资产相关 | 股票类 | Stock\*\*\* | 基本情况 | StockBI\*\*\* | 扩展函数\.Net扩展函数\股票\基本情况 |
| 分红送配 | StockDiv\*\*\* | 扩展函数\.Net扩展函数\股票\分红送股 |
| 高层持股 | StockCLSH\*\*\* | 扩展函数\.Net扩展函数\股票\高层持股 |
| 股本结构 | StockSS\*\* | 扩展函数\.Net扩展函数\股票\股本结构 |
| 股权分置改革 | StockRSS\*\*\* | 扩展函数\.Net扩展函数\股票\股权分置改革 |
| 机构持股 | StockIS\*\*\* | 扩展函数\.Net扩展函数\股票\机构持股 |
| 募集资金 | StockRF\*\*\* | 扩展函数\.Net扩展函数\股票\募集资金 |
| 十大股东 | StockMS\*\*\* | 扩展函数\.Net扩展函数\股票\十大股东 |
| 财务指标 | StockFA\*\*\* | 扩展函数\.Net扩展函数\财务分析 |
| 基金类 | Fund\*\*\* | 基本情况 | FundBI\*\*\* | 扩展函数\.Net扩展函数\基金\基本情况 |
| 财务分析 | FundFA\*\*\* | 扩展函数\.Net扩展函数\基金\财务分析 |
| 分红拆分 | FundDiv\*\*\* | 扩展函数\.Net扩展函数\基金\分红拆分 |
| 份额结构 | FundSS\*\*\* | 扩展函数\.Net扩展函数\基金\份额结构 |
| 净值 | FundNAW\*\*\* | 扩展函数\.Net扩展函数\基金\基金净值 |
| 风险和收益指标 | FundRR\*\*\* | 扩展函数\.Net扩展函数\基金\基金风险和收益指标 |
| 持股情况 | FundSC\*\*\* | 扩展函数\.Net扩展函数\基金\投资组合\持股情况 |
| 行业配置 | FundIC\*\*\* | 扩展函数\.Net扩展函数\基金\投资组合\行业配置 |
| 资产配置 | FundIR\*\*\* | 扩展函数\.Net扩展函数\基金\投资组合\资产配置 |
| 分级基金 | FundSF\*\*\* | 扩展函数\.Net扩展函数\基金\分级基金 |
| ETF | FundETF\*\*\* | 扩展函数\.Net扩展函数\基金\ETF |
| 债券类 | Bond\*\*\* | 基本情况 | BondBI\*\*\* | 扩展函数\.Net扩展函数\债券\基本情况 |
| 估值 | BondFA\*\*\* | 扩展函数\.Net扩展函数\债券\债券估值 |
| 敏感性指标 | BondSA\*\*\* | 扩展函数\.Net扩展函数\债券\债券敏感性指标 |
| 派息和利率 | BondCI\*\*\* | 扩展函数\.Net扩展函数\债券\债券派息和利率 |
| 收益率 | BondBY\*\*\* | 扩展函数\.Net扩展函数\债券\债券收益率 |
| 债券组合相关指标 | pf\_Bond\*\*\* | 扩展函数\.Net扩展函数\债券\债券组合 |
| 宏观 | HG\_\*\*\* | 宏观 | HG\_\*\*\* | 扩展函数\.Net扩展函数\宏观 |
| 行业 | HY\_\*\*\* | 行业 | HY\_\*\*\* | 扩展函数\.Net扩展函数\行业 |
| 行情 | stock\*\*\* | 行情 | stock\*\*\* | 扩展函数\.Net扩展函数\行情 |
| 期货 | futures\*\*\* | 期货 | futures\*\*\* | 扩展函数\.Net扩展函数\期货 |
| 技术分析 |  | 计算公式 | \*\*\*\_f | 扩展函数\.Net扩展函数\技术分析\\*\*\*\计算公式 |
|  | 返回值 | \*\*\*\_v | 扩展函数\.Net扩展函数\技术分析\\*\*\*\当前值 |
| 板块 | bk\_\*\*\* | 板块 | bk\_\*\*\* | 扩展函数\.Net扩展函数\板块 |
| 权证 | Warrant\*\*\* | 权证 | Warrant\*\*\* | 扩展函数\.Net扩展函数\权证 |
| 资产无关 | 日期相关 |  | 日期 |  | 扩展函数\日期 |
| 数字处理 |  | 数字 |  | 扩展函数\数字 |
| 数组处理 |  | 数组 |  | 扩展函数\数组 |
| 数学方法 |  | 数学方法 |  | 数学方法 |

2、报表函数：报表函数是上层的应用类函数，是提取或计算多个维度的数据，比如二维（股票+指标股票+时间、指标+时间）或三维（股票+指标+时间），主要用于项目展现或生成报表。目录为：公用函数\

|  |  |  |  |
| --- | --- | --- | --- |
| 函数类别I | 函数类别II | 目录 | 命名要求 |
| 股票类 | 单个股票 | 01.个股 | Stock\_\*\*\* |
| 多个股票对比 | 02.股票比较 | Stocks\_\*\*\* |
| 行业板块 | 03.行业板块 | Block\_\*\*\* |
| 基金类 | 单个基金 | 04.基金 | Fund\_\*\*\* |
| 多个基金比较 | 05.基金比较 | Funds\_\*\*\* |
| 基金分类比较 | 06.基金公司比较 | FundBlock\_\*\*\* |
| 债券 | 单个债券 | 07.债券 | Bond\_\*\*\* |
| 多个债券比较 | 08.债券比较 | Bonds\_\*\*\* |
| 策略应用 |  | 10.金融工程 |  |

## 函数升级目录

对于原有模型升级，开发者应当**将原有函数转换为用户函数**，并且在此基础上做升级。这种方式，转化后的用户函数不只函数体和各种系统属性设置与原公用函数相同，路径也相同。即在用户函数相同目录下生成了一个公用函数同名的用户函数。这种方式不需要用户自己创建目录。

# 函数命名

为了后期函数使用的方便性，我们要求函数名要按照规范进行命名，不同类别的函数命名有不同的要求。

1、函数名应以英文字符、数字或下划线组成，不应包含中文字符及其他符号；

2、对不同功能或目录的函数命名要求见3.1[新函数目录](#_新函数目录)；

3、函数名最好以该函数的指标或功能的英文名命名。即考虑第2点后加上指标或功能组成函数名称。

# 函数导出

为了方便模型的上传，开发者提交任务时都应该是**批量导出**的模型包（即带路径的.tslfunc文件）。导出的函数文件名，应是函数别名和日期结合的名字，比如“指数估值2018-12-31.tslfunc”。

# 参数设置

## 系统参数

被执行的函数需要设定相关的系统参数，可以通过函数[函数属性](#_属性设置)设置。执行的时候就会根据所设置的显示到系统参数框内。编辑公用函数的时候一定要注意以下几点：

1. 如果函数和证券、时间、周期、复权等相关，且用户参数没有设置，需依赖系统参数时，必须勾选系统参数；比如close函数，跟当前的证券、时间、周期、复权、基准等相关，我们可以在这个函数的系统设置上看到，该函数都已勾选了这些项。

建立myclosetest函数，返回收盘价数据。做如下尝试，分别把证券代码时间作为系统参数和用户参数，看看在证券数据专家/时间序列数据专家中的使用，会有什么影响。

2、仅作为中间函数存在的，可以不勾选“**不执行默认参数**”，否则必须勾选“不执行默认参数”；

3、如果该函数要用于4个证券数据专家，则需在“所属向导函数”中勾选；

4、如果该函数需要放在左边导航栏中，则需在“指标特性”标签页中勾选“可作为应用执行”，再勾选“所属分类”中的类型。最终函数将放在所属分类目录下。不知道如何操作的查看[.NET使用手册](http://www.tinysoft.com.cn/tsdn/helpdoc/display.tsl?id=15367)-》

## 用户参数

### 用户参数命名

使用英文全称或者英文缩写，与同类型的公用函数参数命名一致。例如输入是代码的参数命名都是StockID，别名为股票代码。如果函数不仅

以下为常见参数命名：

|  |  |  |
| --- | --- | --- |
| 参数名 | 别名 | 类型 |
| StockID | 股票代码 | 证券 |
| FundID | 基金代码 | 证券 |
| BondID | 债券代码 | 证券 |
| FutureID | 期货代码 | 证券 |
| IndexID | 指数代码 | 指数 |
| BegT | 开始日期 | 日期 |
| EndT | 截止日期 | 日期 |
| RDate | 报告期 | 报告期 |

### 用户参数分组

如果用户参数的类型是“用户自定义”，且该参数会用于多个函数，则该参数需要设定用户参数分组功能。在参数设置时，直接在“参数分组”选项中选择就可以。如何使用用户参数分组，请自行查看用户手册。

## 属性设置

### 可作为应用执行

如果函数需要放在左侧导航栏中，则需要再“指标特性”中勾选“可作为应用执行”，并再“所属分类”中勾选对应的导航分类。

### 本地应用执行

如果函数为调用第三方的案例，且没有用rdo2，则需勾选“本地应用执行”

### 可作为选择条件

如果函数需要放在四个证券数据专家中，则需勾选“可作为选择条件”，并再“所属分类”中勾选对应的分类。另外，如果有代码日期参数的，还需要注意代码和日期参数的设定，是通过系统参数设置还是通过用户参数设置。

### 可作为返回内容

### 不执行默认参数

如果函数仅作为子函数使用，不提供用户直接执行的函数，可以不勾选“不执行默认参数”，否则需要勾选。

# 函数说明

每个函数都应该有函数别名、简单的函数说明以及详细的TSDOC（文档注释），这样有利于使用文档是生成和函数目录管理。

## 函数别名

函数别名即该函数的中文名称，需简单明了。用户可以通过函数别名搜索到该函数。

## 函数说明

函数说明一般在函数体开头就编辑，对该函数的功能及使用需注意的事项进行简单说明。在函数语句中，也应对关键的步骤写一些注释，便于他人管理和升级。

## TSDOC（文档注释）

在编写函数时，要注意并重视注释的编辑。增加注释不仅提高函数的可读性，同时也是为了代码管理的继承性。

规范编辑函数的注释，按照注释的关键字和关键词进行编辑，天软将可以识别并直接生成相应的说明文档。

### 文档注释块

{\*\*

\*\*}或者

(\*\*

\*\*)

注意开始的\*\*后面有空格结尾的 \*\*前面有空格

### 关键词

关键词采用函数的形式,目前支持的关键词如下

@param(参数1)(参数2)说明

@return(参数)说明

@explan(参数)说明

@example(参数)说明

多个关键词在一个注释块中采用 %%分割,由于 @expample内容太多,将@expample作为一个单独的块

关键词后面接括号每个关键词序号的括号不同,如果括号中的参数有( 请加转义符，即 \(

### 代码范例

type ca = class

\_filed1 ;

{\*\*

@explan(说明) 一个测试%%

@param(\_filed1)(integer) 成员变量

\*\*}

(\*\*

@example(范例1)

oa := new ca(3);

echo oa.inc(5);

\*\*)

functioninc(v);

begin

(\*\*

@param(v)(integer) 被加数%%

@return(integ) 返回值

\*\*)

return v+\_filed1;

end

function create(p)

begin

if ifnumber(p) then

\_filed1 := p;

else p := 0;

end

end

# 函数效率

作为公用函数被使用，开发者应当十分注重函数的效率问题，用户也许在一个项目里就调用了该公用函数几亿甚至几十亿次，开发者设计的模型减少一个没有必要的算符操作，就有可能为用户的整体效率提高许多，这里原有模型升级和新模型的开发都必须注重模型的效率。以下是常见的一些提高模型效率操作：

## 设置和备份系统参数

应该清楚系统参数保存与恢复，有时候是不必要设置的，慎用BackUpSystemParameters2，该函数是把所有的系统函数保存然后恢复，一般的模型中，可能只是修改一个系统参数，应该用with \*代替，或者只保存（恢复）一个系统参数。

例如公用模型IsStockTradeDay：

Function IsStockTradeDay(EndT);

Begin

//++++++++++++++++++++++++++++++++++++++++++++++++++++//

//指定日股票是否进行了交易

//++++++++++++++++++++++++++++++++++++++++++++++++++++//

oV:=BackUpSystemParameters();

SetSysParam(PN\_Cycle(),cy\_day());

v:=IsTradeDay(EndT);

RestoreSystemParameters(oV);

return v;

End;

这里可以这样：

Function IsStockTradeDay(EndT);

Begin

//++++++++++++++++++++++++++++++++++++++++++++++++++++//

//指定日股票是否进行了交易

//++++++++++++++++++++++++++++++++++++++++++++++++++++//

With \*,Array(PN\_Cycle():cy\_day()) Do

v:=IsTradeDay(EndT);

return v;

End;

还可以这样（只保存和恢复一个系统参数）：

Function IsStockTradeDay(EndT);

Begin

//++++++++++++++++++++++++++++++++++++++++++++++++++++//

//指定日股票是否进行了交易

//++++++++++++++++++++++++++++++++++++++++++++++++++++//

SYS\_CYCEL:=GetSysParam(Pn\_Cycle());

v:=IsTradeDay(EndT);

Setsysparam(PN\_Cycle(),SYS\_CYCEL);

return v;

End;

## 多余计算

此类问题常见于循环和select语句中，开发者应该十分清楚，哪些必须在循环里计算，哪些是不必要不合理的。比如设置系统周期setsysparam，对股票循环的话，这个设置应该在循环外设置就可以。

例如以下代码：

StockArr:=Getbk('A股');

R:=Array();

For nI:=0 to Length(StockArr)-1 do

Begin

SetSysparam(Pn\_stock(),StockArr[nI]);

Setsysparam(Pn\_cycle(),Cy\_day());

SetSysParam(Pn\_date(),20180115T);

R[nI]:=Close();

End;

Return R;

此段代码应该修改为：

StockArr:=Getbk('A股');

R:=Array();

Setsysparam(Pn\_cycle(),Cy\_day());

SetSysParam(Pn\_date(),20180115T);

For nI:=0 to Length(StockArr)-1 do

Begin

SetSysparam(Pn\_stock(),StockArr[nI]);

R[nI]:=Close();

End;

Return R;

更多的情况在select语句里面where子句后面，例如下面代码：

Select \* From Data Where ['日期']=Inttodate(EndT) End;

如果表格Data长度有一万，那就是多余做了9999次Inttodate计算，这里应该：

T:=Inttodate(EndT);

Select \* From Data Where ['日期']=T End;

* 可以通过优化功能，查看花费最长时间或调用最多次数的函数或计算，看看是否调用的次数是否合理，花费的时间是否合理，不合理的话怎么找到替换的方法。
* 必须要考虑是否调用合理

## 熟悉公用函数

开发中，有些功能需要调用其他公用函数，比如求股票的Alpha，显然可以直接调用公用函数StockBetaAlpha2，但是这个函数是从取数据到计算alpha，beta，VaR等N多指标全能函数，如果你的模型中本身已经取得了计算这些指标的数据，何不直接使用公式进行求解？很多公用函数是为了满足某些需求开发的，在某些环境下使用，但并不一定是哪里都适用。所以，用的时候要清楚。

## 其他代码优化技巧

类似取数据，减少不必要的调用和计算（见7.2多余计算），是否需要做缓存，是否开发支持网格功能的版本等，请开发者阅读相关的专题文档，细致入微的开发当能无可挑剔。

# 内容规范

## 代码对齐

* 代码对齐、缩进，一般使用tab格(一般为3格)缩进
* Select 太长不好阅读，列字段时可以过行

可以体验一下：

不过行的代码，不好读：

return select ["StockID"] as "代码",["StockName"] as "名称",DateTimeToStr(["date"]) as "时间", ["price"] as "价格", ["vol"] as "成交量", ["amount"] as "成交金额", ["cjbs"] as "成交笔数", ["yclose"] as "上次价", ["syl1"] as "市盈率1", ["syl2"] as "市盈率2", ["buy1"] as "买一价", ["buy2"] as "买二价", ["buy3"] as "买三价", ["buy4"] as "买四价", ["buy5"] as "买五价" from tradetable datekey BegT to EndT+0.99999 of DefaultStockID() end;

过行的代码，可读性强：

return select ["StockID"] as "代码",

["StockName"] as "名称",

DateTimeToStr(["date"]) as "时间",

["price"] as "价格",

["vol"] as "成交量",

["amount"] as "成交金额",

["cjbs"] as "成交笔数",

["yclose"] as "上次价",

["syl1"] as "市盈率1",

["syl2"] as "市盈率2",

["buy1"] as "买一价",

["buy2"] as "买二价",

["buy3"] as "买三价",

["buy4"] as "买四价",

["buy5"] as "买五价"

from tradetable datekey BegT to EndT+0.99999 of DefaultStockID() end;

## 变量

变量命名规范，减少r1/r2/t1/t2/a1/a2等的使用，尽量使用与数据或功能相关的变量名，[参考金融相关英文词汇](#_附：金融相关英文词汇)。变量的命名一般由字符和数字组成。

## 一个函数实现一个功能

尽量一个函数实现一个功能或计算一个指标。比如有一个需求，计算全市场一段时间的指标A。指标A的功能应先用一个函数fun1实现，另一个函数fun2实现一个股票一段时间的指标A，即fun2中时间循环调用fun1。最后，fun3对全市场股票循环，调用fun2。当然，时间循环和股票循环可以调换，先后关系看需求。

对于更复杂的需求，比如要计算多个指标或需要对数据进行各种处理。同样的，也需要尽量分解为小函数或方法，一个函数或方法对应实现一个指标或处理一个数据问题，再由上层函数调用。

## 表ID和字段ID

**表ID**

由于天软的函数需要用在不同的数据源中，不止是天软的数据，客户的服务器可能接的是第三方的数据，比如wind、聚源、财汇等。之前的数据处理中，没有直接在清洗数据层考虑统一规范的问题。故我们需要在函数层考虑。比如，利润分配表，天软的表ID，是46，即在天软客户端，如果使用的是天软数据，那么可以通过infoarray(46)来获取利润表的数据。但是，如果客户那边的数据是wind的，wind入天软服务器时，这个表的ID是752，所以使用wind数据的天软服务器，提取利润表应该是infoarray(752)。而我们开发的公用函数应该是要用在各种数据提供商上的，所以，需要使用公共表ID。函数DatabaseInfoArrayId(公共ID) 即实现了这个功能。比如95即利润表的公共ID，DatabaseInfoArrayId(95)返回的，就是对应不同数据商的不同ID。具体的公用ID的对应，可以直接看这个函数。

比如公用函数IsST\_3判断指定日是否ST，代码如下：

t:=InfoArrayExt(DatabaseInfoArrayId(33),SCN\_ChangeDateField(),DateToInt(EndT),-1);

if isTable(t) then

begin

v2:=t[0][SCN\_AfterNameField()];

return Pos('ST',v2)>0 or Pos('\*',v2)>0 or Pos('ＳＴ',v2)>0;

end

else

return False;

该函数需要取名称变更表的数据进行判断，名称变更表ID在天软是 14 ，但这里公用函数，需要考虑其他数据商不同ID的问题，故用公共ID 33 ，通过函数DatabaseInfoArrayId(33) 可以获得对应数据商下的实际ID，再进行数据提取判断。

**字段ID**

同样的，字段ID也需要进行规范。比如使用天软数据时，46002是营业收入，report(46002,rdate)可以取到某报告期的营业收入，但是，如果使用的是第三方的数据，就不能直接使用46002了。字段ID跟表ID一样有一个公共ID对照的函数DataBaseRatioId。使用方法相同，自行在函数中查看公共ID对照。

比如公用函数stockpe的实现：

if ifnil(DataType) or DataType=0 then

//归属于母公司所有者净利润对应ID

DBR := DataBaseRatioId(9);

else

//扣除非经常性损益后的净利润对应ID

DBR :=DataBaseRatioId(19);

//最新年度报告期

RDate:=NewReportDateOfYear2(EndT);

//总市值

v1:=StockTotalValue(EndT)\*10000.;

//最新年度报告净利润/扣除非经常性损益后的净利润

v2:=ReportNew(DBR,RDate);

return v1/v2;

函数中即使用了公共ID 9和19，在天软数据中实际对应归属于母公司所有者净利润46078和扣除非经常性损益后的净利润42017。

**字段名**

字段名同样也要规范。比如“暂停、中止、恢复上市”表，对于股票的变更类型“暂停上市”，在天软中的字段为“暂停上市”，在财汇中为“被暂停上市日”，由于这种数据处理的非标准化，我们也有一套字段对照，见目录“扩展函数-》基本面-》数据库提供商-》数据表字段对照”，该目录下所有函数均为字段对照函数。

比如A股总股本函数TSharesOfA，代码如下：

return GetNShares(InfoId,EndT,SSS\_TotalSharesField(),SSS\_ChangeDateField(),SSS\_DeclaredDateField());

由于股本结构表中，不同数据商的字段不同，为了取到对应的变动日后的总股本，变动日在不同的数据商中说法可能不同。比如天软数据中，就叫“变动日”，而Wind使用的名称是“变动日期(除权日)”，我们通过函数SSS\_ChangeDateField()做了对照。在使用时，比如上面那段代码，就直接用了这个对照函数获得对应的字段名，从而取相应的数据判断。

# 升级日志

提交升级函数的时候，需同时提交相应的升级日志，说明升级的原因和本次升级涉及的函数。比如：

|  |  |  |
| --- | --- | --- |
| **No** | **修改说明** | **操作步骤** |
| 01 | 3-13日的股本函数修正没有考虑兼容历史模型，故修改函数：  1、GetNShares，系统参数PN\_ChangedDate()=0或不设置时，获取股本数据会考虑变动日+公布日；PN\_ChangedDate()=1时，获取股本数据只考虑变动日（历史模型）。  2、StockShares，同上。  3、增加系统函数PN\_ChangedDate() | 批量导入"股本结构2019-4-11.tslfunc"，文件中包含GetNShares和StockShares两个函数。 |

# 附：金融相关英文词汇

## 交易所和机构

上交所 Shanghai Exchange

A股市场 A-share market

B股市场 B-share market

证券交易所 bourse

保证金融资账户 margin finance account

货币市场 money market

熊市 bear market

牛市 bull market

基金托管银行 fund custodian bank

基金经理/管理公司 fund manager

券商 broker / dealer

## 证券

大盘股 big macs , big-cap stock, large-cap stock

股票/股权 stock ，share，equity

商业票据 commercial paper

国库券 Treasuries

国债 government bond

国债期货 treasury bond futures

投资组合 portfolio

期权合约 option contract

开方式基金 open-end fund

封闭式基金 closed-end fund

债券 bond, debts

纳斯达克指数 NASDAQ

日经指数 Nikkei Index

标准普尔500指数 the Standard and Poor’s 500

绩优股 blue chip

红股 bonus share

成分指数 component index

可转换债券 convertible bond

企业债券 corporate bond

债权转股权 debt-for-equity swap

工业股票 industrial stock

高科技板块 high-tech sector

红筹股 red chips

可流通股份 negotiable share

大盘 the big board

存托凭证 Depository Receipt （DR）

美国存托凭证 American Depository Receipt (ADR)

全球存托凭证 Global Depository Receipt (GDR)

中国存托凭证 Chinese Depository Receipt (CDR)

金融衍生产品 financial derivatives

## 交易

早盘 morning session

午盘 afternoon session

开始前时段 pre-opening session

上市 offering, list

退市/摘牌 delist

交易 trading

停板 trading halt

停牌 suspension，suspended trading

成交额 turnover

成交量 trading volume

权重 weighting

收益 return

利润 profit

看涨的 bullish

上涨 gain

下跌 decline,fall, slip, retreat

做价 kiting

动态对冲 dynamic hedging

参考平衡成交量 Indicative Equilibrium Volume (IEV)

参考平衡价格 Indicative Equilibrium Price (IEP)

参考发行价 Indicated issue price

发行价 issue price

佣金 commission

存货 inventory

期权 option

看涨期权 call option

看跌期权 put option

等价期权 at-the-money option

套利 arbitrage

结算货币 transaction currency

期货交易 forward exchange

即期交易 spot transaction

远期交易 forward transaction

择期交易 option forward transaction

掉期交易 swap transaction

掉期 swap

报价 quote

交割 settlement and delivery

买入价 buying rate

卖出价 selling rate

平价 at par

升水 premium

贴水 discount

股息 dividend

本币 domestic currency

浮动利率 floating rate

首次公开发行 IPO initial public offering

杠杆 leverage

敏感度分析 sensitivity analysis

清算,结算 clearing

清算价 clearing price

最大差价限额 maximum spread

最低交易数额 minimum transaction size

最低价格变动 minimum price fluctuation , tick

最佳成交价 best execution price

最后成交价 last recorded price, last traded price

期权定价 option valuation

期权定价模式 option pricing model

期权金 premium

期权持仓 option position

贴现 discount

贴现率 discount rate

创历史新高 record high

市值 market capitalization

## 财务

商誉 goodwill

风险资本 venture capital

账面价值 book value

实际资本 physical capital

控股股东 controlling shareholder

票面利率 coupon, coupon rate

票面值 par value

速动资金 liquid capital

速动资产 liquid asset

期末存货 closing stock

期初存货 opening stock

短期资金 short-term money

短期借款 short term borrowing

核心/非核心资产 core assets / non-core assets

虚假债务 deceptive accounting

金融重组 financial reorganization

财务报表 financial statement

损益表 income statement

市盈率 P/E price/earning ratio

## 宏观

宏观经济 macro economic

国内生产总值 gross domestic product (GDP)

国民生产总值 gross national product (GNP)

## 其他

无风险利率 risk-free interest rate

无风险对冲 delta hedge

税后利润 after-tax profits

稀释 dilution

程式交易 programme trading