

缠论插件使用手册

V2. 1

草帽路飞

goodpp2012@gmail . com

2011年01月

| | |
|--------------------------|----|
| 2011 年 01 月..... | 1 |
| 1 插件安装..... | 4 |
| 1.1 下载地址..... | 4 |
| 1.2 安装环境..... | 4 |
| 1.3 大智慧下安装..... | 4 |
| 1.4 飞狐下安装..... | 5 |
| 2 公式使用..... | 6 |
| 2.1 版本说明..... | 6 |
| 2.2 公式说明..... | 6 |
| 2.3 函数调用..... | 7 |
| 2.4 使用示例..... | 7 |
| 2.5 替换悟多..... | 10 |
| 2.6 一些技巧..... | 11 |
| 2.6.1 前 N 个笔高低点的取值..... | 11 |
| 2.6.2 计算一笔间包含处理后的距离..... | 11 |
| 2.7 这个给有兴趣的筒子..... | 11 |
| 3 函数说明..... | 17 |
| 3.1 函数列表..... | 17 |
| 3.1.1 函数名大写..... | 17 |
| 3.1.2 函数名小写..... | 17 |
| 3.2 详细函数说明(大写)..... | 18 |
| 3.2.1 VERSION | 18 |
| 3.2.2 DATE | 19 |
| 3.2.3 JXZT | 19 |
| 3.2.4 JXYL..... | 19 |
| 3.2.5 JXZC | 19 |
| 3.2.6 JXYL2..... | 20 |
| 3.2.7 JXZC2 | 20 |
| 3.3 详细函数说明(小写)..... | 20 |
| 3.3.1 kxian..... | 20 |
| 3.3.2 kxg..... | 21 |
| 3.3.3 kxd..... | 21 |
| 3.3.4 kxfx | 21 |
| 3.3.5 kxno..... | 22 |
| 3.3.6 kxnobh..... | 22 |
| 3.3.7 kxqj..... | 22 |
| 3.3.8 kxqk..... | 23 |
| 3.3.9 bi | 23 |
| 3.3.10 bilast..... | 23 |
| 3.3.11 bigglast | 23 |
| 3.3.12 biddlast | 24 |
| 3.3.13 duan..... | 24 |
| 3.3.14 duanlast | 24 |

| | |
|-------------------------|----|
| 3.3.15 duangglast | 24 |
| 3.3.16 duanddlast | 25 |
| 3.3.17 zs | 25 |
| 3.3.18 zszg..... | 25 |
| 3.3.19 zszd..... | 25 |
| 3.3.20 zsgg | 26 |
| 3.3.21 zsdd | 26 |
| 3.3.22 zszz..... | 26 |
| 3.3.23 zskslast | 26 |
| 3.3.24 zsjslast | 27 |
| 3.4 函数使用注意事项..... | 27 |

前言

本人对缠论一见钟情，兴趣所致，才写此插件，同时也是把这当成一种学习方式，效果不错咯。插件不会有任何使用限制。本插件从设计到写代码，写文档都是本人一个人利用业余时间花了一个半月时间完成的。难免会存在 BUG，错误等等。再加上本人学禅时日尚短，难免有些理解有错误。各位缠论学长如果有理解错误或者 BUG 的地方，可以通过 EMAIL 或在我的 Blog 和我交流，请各位学长多多指教。

本人的联系方式只会有邮件和 BLOG 这两种，不会留 QQ 不会建群。现在骗子多还是提醒各位筒子不要上当。

Email: goodpp2012@gmail.com

Blog: <http://blog.sina.com.cn/strawhatkids>

1 插件安装

1.1 下载地址

本人 Blog 置顶贴 http://blog.sina.com.cn/s/blog_71e0c1fc0100opcr.html

在置顶贴中下载，请自行校对 MD5。

1.2 安装环境

本插件支持大智慧和飞狐两款软件。具体那些版本适用我也不是清楚-_-\\。我是用分析家 3.1 的接口开发的，大智慧新一代和飞狐 5 以上的版本肯定是支持的。

1.3 大智慧下安装

- 1) 拷贝 chanlun.dll 至大智慧的安装目录
- 2) 导入相关公式，公式->公式引入



1.4 飞狐下安装

- 1) 拷贝 chanlun.dll 至飞狐安装目录下的 fml.dll 文件夹中
- 2) 导入相关公式



选择相关 alg 文件后导入



2 公式使用

2.1 版本说明

V2.1 版是我写的第三个版本,也是第一个在网上公开发布的版本。意图解决以下 2 个问题:

- 1) 自动画线。
- 2) 把缠论形态学里相关的数据都封装成了对应的函数。方便大家调用,以方便设计自己的公式。

画的不对的地方请学长们拍砖。封装的不好的地方请学长们批评。

2.2 公式说明

公式分为大智慧和飞狐两个版本。除了面积 MACD, 其他三个公式都是为了演示本插件的用法, 基本演示了本插件的所有函数。

2.3 函数调用

在公式中使用“chanlun@函数名”的方式调用插件中的函数。函数名大写的函数为可单独调用的函数，函数名小写的函数为不可单独调用的函数(单独调用得不到数据)。具体请参考函数说明。

例：

```
kx:="chanlun@kxian";

kxg:="chanlun@kxg";

kxd:="chanlun@kxd";

kxfx:="chanlun@kxfx";

kxno:"chanlun@kxno", linethick0, coloryellow;

kxnobh:"chanlun@kxnobh", linethick0, coloryellow;

笔:="chanlun@bi";
```

2.4 使用示例

```
{示例开始: }

kx:="chanlun@kxian"; {这个放第一个 要不会出问题滴_-\~}

kxg:="chanlun@kxg";

kxd:="chanlun@kxd";

kxfx:="chanlun@kxfx";

kxno:"chanlun@kxno", linethick0, coloryellow;

kxnobh:"chanlun@kxnobh", linethick0, coloryellow;

笔:="chanlun@bi";

bilast:"chanlun@bilast", linethick0, coloryellow;

段:="chanlun@duan";

duanlast:"chanlun@duanlast", linethick0, coloryellow;

中枢:="chanlun@zs";

zszg:="chanlun@zszg";

zszd:="chanlun@zszd";

zsgg:="chanlun@zsgg";

zsdd:="chanlun@zsdd";

zszz:="chanlun@zszz";

zsks:="chanlun@zskslast";

zsjs:="chanlun@zsjslast";

bifx:=-REF(笔, bilast);
```

```

duanfx:=-REF(段,duanlast);

kxqj:="chanlun@kxqj";
fxqj:ref(kxqj,bilast), linethick0, coloryellow;

{TT:"chanlun@TT", linethick0, colorwhite;}

{画中枢Begin}
{以背景方式画}
{
STICKLINE(zszg>zszd AND duanfx=1, zszg, zszd, 20, 502), colorf00ff0;
STICKLINE(zszg>zszd AND duanfx=-1, zszg, zszd, 20, 502), colore6002d;
}
{以直线方式画}
{上涨中枢}
DRAWLINE(zsks=1 AND duanfx=1, zszg, zsjs=1 AND duanfx=1, zszg, 0), colormagenta, linethick2;
DRAWLINE(zsks=1 AND duanfx=1, zszd, zsjs=1 AND duanfx=1, zszd, 0), colormagenta, linethick2;
STICKLINE(zsks=1 AND duanfx=1, zszg, zszd, 1 AND duanfx=1, -501), colormagenta, linethick2;
STICKLINE(zsjs=1 AND duanfx=1, zszg, zszd, 1 AND duanfx=1, 501), colormagenta, linethick2;
{下跌中枢}
DRAWLINE(zsks=1 AND duanfx=-1, zszg, zsjs=1 AND duanfx=-1, zszg, 0), colorCyan, linethick2;
DRAWLINE(zsks=1 AND duanfx=-1, zszd, zsjs=1 AND duanfx=-1, zszd, 0), colorCyan, linethick2;
STICKLINE(zsks=1 AND duanfx=-1, zszg, zszd, 1 AND duanfx=-1, -501), colorCyan, linethick2;
STICKLINE(zsjs=1 AND duanfx=-1, zszg, zszd, 1 AND duanfx=-1, 501), colorCyan, linethick2;
{中枢震荡监视器}
DRAWLINE(zsks=1, zszz, zsjs=1, zszz, 0), coloryellow;
{画中枢END}

{显示缺口BEGIN}
{只显示5F及5F级别以上图的缺口}
kxqk:="chanlun@kxqk";
{最后未封闭的最后一个上涨（下跌）缺口到现在的周期}
kxqkup:BARSLAST(kxqk=1), linethick0, colorMagenta;
kxqkdn:BARSLAST(kxqk=-1), linethick0, colorGreen;
{缺口}
DRAWTEXT(kxqk=1, l*0.992, '☆'), colorMagenta, align1 ; {未封闭上涨缺口}
DRAWTEXT(kxqk=2, l*0.992, '★'), colorMagenta, align1 ; {已封闭上涨缺口}
DRAWTEXT(kxqk=-1, h*1.008, '☆'), colorGreen, align1, valign2; {未封闭下跌缺口}
DRAWTEXT(kxqk=-2, h*1.008, '★'), colorGreen, align1, valign2; {已封闭下跌缺口}
{显示缺口END}

{画K线Begin}
{缠论K线 已经包含处理}
STICKLINE(kxfx>1, kxg, kxd, 2, 1), coloryellow;

```



```

STICKLINE(kxfx<1, kxg, kxd, 2, 1), colorF0F000;
STICKLINE(kxfx=1, kxg, kxd, 2, 1), colorred;
STICKLINE(kxfx=-1, kxg, kxd, 2, 1), colorgreen;
{顶底分型}
STICKLINE(kx>0, kxg, kxd, 5, 1), colorwhite;
STICKLINE(kx<0, kxg, kxd, 5, 1), colorblue;
{笔段}
STICKLINE(笔=1, kxg, kxd, 5, 1), colorgreen, linethick3;
STICKLINE(笔=-1, kxg, kxd, 5, 1), colorred, linethick3;
STICKLINE(段=1, kxg, kxd, 5, 0), colore66878, linethick2;
STICKLINE(段=-1, kxg, kxd, 5, 0), color00cccc, linethick2;
{画K线END}

{画笔段}
drawline(笔=1,h,笔=-1,l,0),coloryellow;
drawline(笔=-1,l,笔=1,h,0),coloryellow;
drawline(段=1,h,段=-1,l,0),colorblue, linethick2;
drawline(段=-1,l,段=1,h,0),colorblue, linethick2;

{笔高低点 周期 力度 提示}
DRAWTEXT(笔=1, h, ''+h+' /T'+bilast+' /'+(h-REF(l,bilast))), coloryellow, align1, valign2;
DRAWTEXT(笔=-1, l, ''+l+' /T'+bilast+' /'+(l-REF(h,bilast))), colorCyan, align1;

{当下提示 周期}
DRAWTEXT(BARSTATUS=2 AND bifax=1 AND bilast>4, h*1.008, 'T'+bilast+' /'+(c-REF(l,bilast))),
colorMagenta, align0, valign2;
DRAWTEXT(BARSTATUS=2 AND bifax=-1 AND bilast>4, l*0.992, 'T'+bilast+' /'+(c-REF(h,bilast))),
colorGreen, align0;

{布林曲线}
{
MID :MA(CLOSE, 26), pointdot, colorwhite;
UPPER:MID + 2*STD(CLOSE, 26), linethick1, coloryellow;
LOWER:MID - 2*STD(CLOSE, 26), linethick1, colorgreen;
收口:=UPPER-LOWER;
partline(UPPER, 收口>ref(收口, 1), rgb(255, 255, 0), 收口<ref(收口, 1), rgb(0, 255, 0)), linethick1,
dotline;
partline(LOWER, 收口>ref(收口, 1), rgb(255, 255, 0), 收口<ref(收口, 1), rgb(0, 255, 0)), linethick1,
dotline;
}

{以下主要演示取笔段中枢数据的方法}
{前N个高点和低点(笔)}
{

```

```

bigglast:="chanlun@bigglast";
biggl1:ref(h, bigglast), linethick0, coloryellow;
biggl2:ref(biggl1, bigglast), linethick0, coloryellow;
biggl3:ref(biggl2, bigglast), linethick0, coloryellow;

biddlast:="chanlun@biddlast";
bidd1:ref(l, biddlast), linethick0, colorcyan;
bidd2:ref(bidd1, biddlast), linethick0, colorcyan;
bidd3:ref(bidd2, biddlast), linethick0, colorcyan;
}

{取前一段的数据}
{
dgglast:="chanlun@duangglast";
dgg1:ref(h, dgglast), linethick0, coloryellow;

dddlast:="chanlun@duanddlast";
ddd1:ref(l, dddlast), linethick0, colorcyan;
}

{取前一个中枢的数据}
{
zszg1:ref(zszg, zsjs), linethick0, coloryellow;
zszd1:ref(zszd, zsjs), linethick0, coloryellow;
};

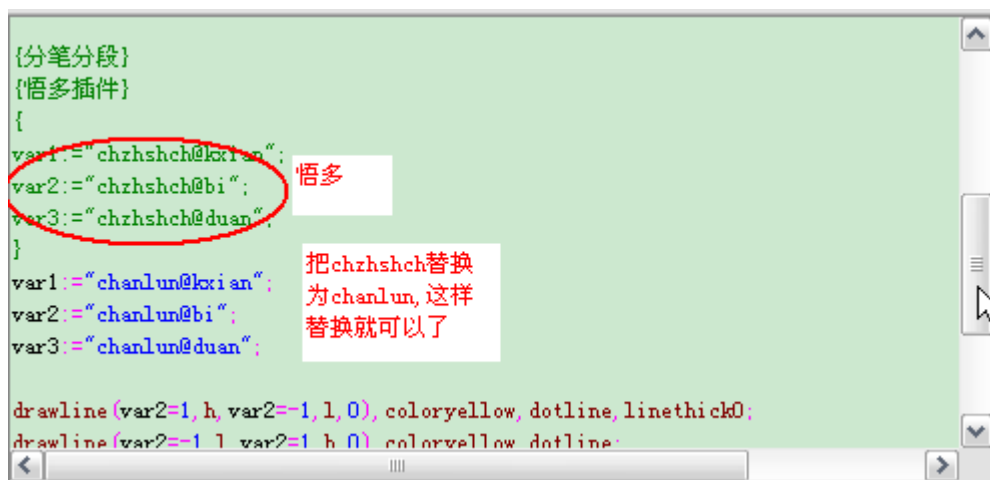
{版本信息Begin}
vsnN0:="chanlun@VERSION";
vsnDate:="chanlun@DATE";
vsnSTR:= ' 版本:' +vsnN0+'.' +vsnDate;
{vsnSTR1:= ' 作者:goodpp2012@gmail.com';}
vsnSTR1:='';
DRAWTEXTREL( 5, 950, vsnSTR+vsnSTR1), colormagenta, linethick2;
{版本信息End};

{示例结束: }

```

2.5 替换悟多

本插件和悟多插件兼容的三个函数为: **kxian**、**bi**、**duan**, 均以1 -1 表示顶底。如果你有相关公式是使用了悟多现在想替换为本插件只需要, 俺下图这样做就可以了。



2.6 一些技巧

这里介绍几个有用的技巧，注意以下代码只是部分。

2.6.1 前 N 个笔高低点的取值

```

{其他省略...}
bigglast:="chanlun@bigglast";
biggl1:ref(h, bigglast), linethick0, coloryellow; {前第一个笔高点}
biggl2:ref(biggl1, bigglast), linethick0, coloryellow; {前第二个笔高点}
biggl3:ref(biggl2, bigglast), linethick0, coloryellow; {前第三个笔高点}

biddlast:="chanlun@biddlast";
biddl1:ref(l, biddlast), linethick0, colorcyan; {前第一个笔低点}
biddl2:ref(biddl1, biddlast), linethick0, colorcyan; {前第二个笔低点}
biddl3:ref(biddl2, biddlast), linethick0, colorcyan; {前第三个笔低点}

```

2.6.2 计算一笔间包含处理后的距离

```

{其他省略...}
kxnobh:="chanlun@kxnobh", linethick0, coloryellow;
包含后笔距离:=kxnobh-REF(kxnobh,bilast);

```

2.7 这个给有兴趣的筒子

```

/*****
****/
/*

```

缠论核心模块 V2.1

单例模式

所有Chanlun接口中缠论核心函数在此模块中实现

@Author 草帽路飞

@Copyright 草帽路飞

@Email goodpp2012@gmail.com

@Blog <http://blog.sina.com.cn/strawhatkids>

@Date 2011.01.01

```
*/
/*****
****/

#ifndef __CHANLUNCORE_H_INCLUDE
#define __CHANLUNCORE_H_INCLUDE

#include "FxjFunc.h"
#include <list>

// 方向
enum FX_STATE
{
    FX_XBH2 = -3, // 前包含后 -3
    FX_XBH,  // 后包含前 -2
    FX_DN,   // 方向向下 -1
    FX_0,    // 0
    FX_UP,   // 方向向上 1
    FX_SBH,  // 后包含前 2
    FX_SBH2  // 前包含后 3
};

// 缺口
enum QK_STATE
{
    QK_N = 0, // 不存在缺口
    QK_Y // 存在缺口
};

//定义基本数据
// 别批评偶不规范啊... 偶只是写着玩 -_-\\~~~
// 缠论K线结构 处理过包含关系的K线
typedef struct tagCHANKX
{
    int m_nNo;           // K线序号 从1开始
    int m_nNobh;         // 经包含处理后的K线序号
```

```

float m_fRealHigh;    // 高值
float m_fRealLow;     // 低值
float m_fHigh;        //包含处理后的高值
float m_fLow;         //包含处理后的低值
int m_nFlag;          //1顶 -1底 0 非顶底
float m_fFxqj;        // 分型区间 如果为顶底 记录区间边界
int m_nFx;            //K线方向 1上 -1下 2 上包含 -2 下包含
int m_nBi;            //笔 1上 -1下 2 上包含 -2 下包含
int m_nDuan;          //段 1上 -1下 2 上包含 -2 下包含
int m_nQk;            //缺口 0没有 1有
} CHANKX;

```

// 笔 (特征序列)

```
typedef struct tagCHANBI
```

```

{
    int m_nNo; // 序号
    int m_nNoh; // 高点K线编号
    int m_nNol; // 低点K线编号
    float m_fHigh; // 高点值
    float m_fLow; // 低点值
    int m_nFx; // 方向 方向 1上 -1下 2 上包含 -2 下包含
    int m_nFlag; // 1顶 -1底
    int m_nQk; // 特征1 2 之间是否存在缺口
} CHANBI;

```

// 段

```
typedef struct tagCHANDUAN
```

```

{
    int m_nNo; // 序号
    int m_nNoh; // 高点K线编号
    int m_nNol; // 低点K线编号
    float m_fHigh; // 高点值
    float m_fLow; // 低点值
    int m_nFlag; // 1顶 -1底
    int m_nZnNum; // 包含特征笔数量 笔=znNum*2 + 1
    int m_nZsNum; // 包含中枢数量
} CHANDUAN;

```

// 走势中枢

```
typedef struct tagCHANZS
```

```

{
    int m_nNo; // 序号
    int m_nDuanNo; // 段序号
    int m_nFlag; // 走势方向 1上 -1下

```

```

    int m_nZsksNo; // 中枢开始所在K线NO
    int m_nZsjsNo; // 中枢结束所在K线NO
    int m_nZnNum; // 包含zn数
    float m_fZszg; // ZG=min(g1、 g2)
    float m_fZszd; // ZD=max(d1、 d2)
    float m_fZsgg; // GG=max(gn);
    float m_fZsdd; // DD=min(dn);
    float m_fZszz; // 震荡中轴(监视器)
} CHANZS;
//定义基本数据 END

// 自定义类型BEGIN
typedef std::list<CHANKX> KXDATA;
typedef std::list<CHANBI> BIDATA;
typedef std::list<CHANDUAN> DUANDATA;
typedef std::list<CHANZS> ZSDATA;

typedef KXDATA::iterator CKXIT;
typedef BIDATA::iterator BIIT;
typedef DUANDATA::iterator DUANIT;
typedef ZSDATA::iterator ZSIT;

typedef KXDATA::const_iterator C_CKXIT;
typedef BIDATA::const_iterator C_BIIT;
typedef DUANDATA::const_iterator C_DUANIT;
typedef ZSDATA::const_iterator C_ZSIT;
// 自定义类型END

//缠K线 分型 笔 段 中枢 算法接口 BEGIN
// 缠K线 算法接口
class ChanlunKxIF
{
private:
    virtual void __DoInitKx(CALCINFO* pData, KXDATA &kxData) = 0;

public:
    void InitKx(CALCINFO* pData, KXDATA &kxData);
};

// 分型 算法接口
class ChanlunFxIF
{
private:
    virtual void __DoInitFx() = 0;

```

```

public:
    void InitFx();
};

// 笔 算法接口
class ChanlunBiIF
{
private:
    virtual void __DoInitBi() = 0;

public:
    void InitBi();
};

// 段 算法接口
class ChanlunDuanIF
{
private:
    virtual void __DoInitDuan() = 0;

public:
    void InitDuan();
};

// 中枢 算法接口
class ChanlunZsIF
{
private:
    virtual void __DoInitZs() = 0;

public:
    void InitZs();
};

// 算法接口 END

// 缠论核心类
class ChanlunCore
{
private:
    ChanlunCore(); // 构造函数
    ~ChanlunCore(); // 析构函数

    static ChanlunCore* ms_pInstance;
};

```

```

ChanlunKxIF* m_pChanlunKx; // 缠K线算法
ChanlunFxIF* m_pChanlunFx; // 顶底分型算法
ChanlunBiIF* m_pChanlunBi; // 笔算法
ChanlunDuanIF* m_pChanlunDuan; // 中枢算法
ChanlunZsIF* m_pChanlunZs; // 中枢算法

KXDATA m_kxData; // 根据缠论处理过包含关系的K线
BIDATA m_xbData; // 向下笔 （向上段的特征序列）
BIDATA m_sbData; // 向上笔 （向下段的特征序列）
DUANDATA m_dData; // 段
ZSDATA m_zsData; // 中枢

void __InitData(CALCINFO* pData); // 初始化数据

public:
    static ChanlunCore* GetInstance(); //获取 唯一实例 单例模式

    void InitKx(CALCINFO* pData); // 初始化缠论K线
    void InitFX(); // 初始化分型
    void InitBi(); // 初始化笔
    void InitDuan(); // 初始化段
    void InitZhongshu(); // 初始化中枢

    // Getter
    const KXDATA GetCkxData();
    const BIDATA GetXbData();
    const BIDATA GetSbData();
    const DUANDATA GetDuanData();
    const ZSDATA GetZsData();
};
// 缠论核心类END

#endif // __CHANLUNTOOLS_H_INCLUDE

```


3 函数说明

3.1 函数列表

3.1.1 函数名大写

| 函数名 | 函数说明 | 依赖关系 | 加入版本 |
|---------|------|------|------------|
| VERSION | 版本 | 无 | @since1.01 |
| DATE | 发布日期 | 无 | @since1.01 |
| JXZT | 均线状态 | 无 | @since1.01 |
| JXYL | 均线压力 | 无 | @since1.01 |
| JXZC | 均线支撑 | 无 | @since1.01 |
| JXYL2 | 均线压力 | 无 | @since1.02 |
| JXZC2 | 均线支撑 | 无 | @since1.02 |

3.1.2 函数名小写

| 函数名 | 函数说明 | 依赖关系 | 加入版本 |
|----------|-------------------|-------|------------|
| kxian | 初始化缠论 K 线 | 无 | @since1.01 |
| kxg | 缠论 K 线最高值(经包含处理) | kxian | @since1.01 |
| kxd | 缠论 K 线最底值(经包含处理) | kxian | @since1.01 |
| kxfx | 缠论 K 线方向 | kxian | @since1.01 |
| kxno | K 线序号(唯一) | kxian | @since2.01 |
| kxnobh | 经包含处理后的 K 线序号 | kxian | @since2.01 |
| kxqj | K 线为顶底时的 K 线区间边界值 | kxian | @since2.01 |
| kxqk | 是否存在缺口 | kxian | @since2.01 |
| | | | |
| bi | 笔 | kxian | @since1.01 |
| bilast | 距离上一笔结束的时间周期 | bi | @since1.01 |
| bigglast | 距离上 1 个笔高点的时间周期 | bi | @since1.02 |

| | | | |
|------------|-----------------|------|------------|
| biddlast | 距离上 1 个笔低点的时间周期 | bi | @since1.02 |
| | | | |
| duan | 段 | bi | @since1.01 |
| duanlast | 距离上一段结束的时间周期 | duan | @since1.01 |
| duangglast | 距离上 1 个段高点的时间周期 | duan | @since1.02 |
| duanddlast | 距离上 1 个段低点的时间周期 | duan | @since1.02 |
| | | | |
| zs | 走势中枢 | duan | @since1.02 |
| zszg | 中枢 ZG | zs | @since1.02 |
| zszd | 中枢 ZD | zs | @since1.02 |
| zsgg | 中枢 GG | zs | @since1.02 |
| zsdd | 中枢 DD | zs | @since1.02 |
| zszz | 中枢中轴 | zs | @since1.02 |
| zskslast | 距离上一个中枢开始的时间周期 | zs | @since1.02 |
| zsjslast | 距离上一个中枢结束的时间周期 | zs | @since1.02 |

3.2 详细函数说明(大写)

3.2.1 VERSION

函数名: VERSION

参数: 无

since: 1.01

说明: 软件版本号, 每个版本都向前兼容以前版本。

如: 2.012

2 主版本号 01 小版本号 2 发行版本标识

发行版本标识: 1 alpha(内测版) 2 beta(公测版) 3 standard 标准版 4-5 标准修正版

9 release final 最终版(不再更新该版本)

2.012 即为 2.01beta

示例: vsnN0:=“chan1un@VERSION”;

3.2.2 DATE

函数名: DATE

参数: 无

since: 1.01

说明: 发行日期, 一般和版本号结合, 如 2.011.2011.0101

示例: `vsndate:~"chanlun@DATE";`

3.2.3 JXZT

函数名: JXZT

参数: 无

since: 1.01

说明: 表示的是当前收盘价在均线中所处的状态,

返回值: 0-8

0 表示当前在所有 1-8 类缠论均线之下 8 表示在所有均线之上

示例: 均线状态:~"chanlun@JXZT",coloryellow;

3.2.4 JXYL

函数名: JXYL

参数: 无

since: 1.01

说明: 表示的是当前收盘价最近的压力均线类型

返回值: 1-9

1-8 分别对应缠论里的 8 类均线 1=5, 2=13, 3=21, 4=34, 5=55, 6=89, 7=144, 8=233

9 表示为无压力均线

3.2.5 JXZC

函数名: JXZC

参数：无

since: 1.01

说明：表示的是当前收盘价最近的支撑均线类型,

返回值：0-8

1-8 分别对应缠论里的 8 类均线 1=5, 2=13, 3=21, 4=34, 5=55, 6=89, 7=144, 8=233

0 表示为无支撑均线

3.2.6 JXYL2

函数名：JXYL2

参数：无

since: 1.02

说明：表示的是当前收盘价最近的压力均线。

返回值：5 13 21 34 55 89 144 233

3.2.7 JXZC2

函数名：JXZC2

参数：无

since: 1.02

说明：表示的是当前收盘价最近的支撑均线。

返回值：5 13 21 34 55 89 144 233

3.3 详细函数说明(小写)

3.3.1 kxian

函数名：kxian

参数：无

依赖：无

since: 1.01

说明：初始化数据，得到经过包含等关系处理后的缠论 K 线，并返回顶底分型。兼容悟多。

kxian 是所有小写函数都需要依赖的函数放在最前面调用。

返回值：-2-2

-1 底分型 1 顶分型 0 普通 K 线

-2 下跌中底分接底分，被废弃的低分。2 上涨中顶分接顶分，被废弃的顶分。

3.3.2 kxg

函数名：kxg

参数：无

依赖：kxian

since：1.01

说明：返回经过包含处理后的 K 线高值。

3.3.3 kxd

函数名：kxd

参数：无

依赖：kxian

since：1.01

说明：返回经过包含处理后的 K 线低值。

3.3.4 kxfx

函数名：kxfx

参数：无

依赖：kxian

since：1.01

说明：返回经过包含处理后的 K 线方向。

返回值：-3-3

-1 下跌 无包含关系, -2 下跌后 K 包含前 K, -3 下跌前 K 包含后 K。

1 上涨 无包含关系, 2 上涨后 K 包含前 K, 3 上涨前 K 包含后 K。

3.3.5 kxno

函数名: kxno

参数: 无

依赖: kxian

since: 2.01

说明: K 线序号, 唯一值, 从 1 开始

3.3.6 kxnobh

函数名: kxnobh

参数: 无

依赖: kxian

since: 2.01

说明: 处理包含关系后的 K 线序号, 如果几根 K 线的该值相同, 表示这些 K 线经过包含处理后为同一根 K 线。

3.3.7 kxqj

函数名: kxqj

参数: 无

依赖: kxian

since: 2.01

说明: K 线区间, 顶底分型 K 线最高(最低)K 线所在的区间

返回值: 如果为顶 该值为顶 K 经过包含关系处理后的低值

如果为底, 该值为底 K 经过包含关系处理后的高值

其他情况, 返回-1

3.3.8 kxqk

函数名: kxqk

参数: 无

依赖: kxian

since: 2.01

说明: K 线缺口, 该 K 线与上一根 K 线之间是否存在缺口。只对 5F 及 5F 以上图有效。

返回值: 0 无 1 未封闭上涨缺口 2 已封闭上涨缺口 -1 未封闭下跌缺口-2 已封闭下跌缺口

3.3.9 bi

函数名: bi

参数: 无

依赖: kxian

since: 1.01

说明: 笔处理, 兼容悟多。

返回值: 0 普通 K 线 1 笔顶点 -1 笔底点 2 废弃的笔顶点 -2 废弃的笔底点

3.3.10 bilast

函数名: bilast

参数: 无

依赖: bi

since: 1.01

说明: 距离上一笔顶(底)点结束的时间周期, 前顶(底)点不存在时返回-1。

3.3.11 bigglast

函数名: bigglast

参数: 无

依赖: bi

since: 1.02

说明：距离上一笔顶点结束的时间周期，前顶点不存在时返回-1。

3.3.12 biddlast

函数名: biddlast

参数: 无

依赖: bi

since: 1.02

说明：距离上一笔底点结束的时间周期，前底点不存在时返回-1。

3.3.13 duan

函数名: duan

参数: 无

依赖: bi

since: 1.02

说明：段处理，兼容悟多。

返回值: 0 普通 K 线 1 段顶点 -1 段底点 2 废弃的段顶点 -2 废弃的段底点

3.3.14 duanlast

函数名: duanlast

参数: 无

依赖: duan

since: 1.02

说明：距离上一段顶(底)点结束的时间周期,, 前顶(底)点不存在时返回-1。

3.3.15 duangglast

函数名: duangglast

参数：无

依赖：duan

since：1.02

说明：距离上一段顶点结束的时间周期，前顶点不存在时返回-1。

3.3.16 duanddlast

函数名：duanddlast

参数：无

依赖：duan

since：1.02

说明：距离上一段底点结束的时间周期，前底点不存在时返回-1。

3.3.17 zs

函数名：zs

参数：无

依赖：duan

since：1.02

说明：走势中枢处理，返回值-1

3.3.18 zszg

函数名：zszg

参数：无

依赖：zs

since：1.02

说明：中枢 ZG 值，非构成中枢的 K 线该值返回 0。

3.3.19 zszd

函数名：zszd

参数：无

依赖：zs

since：1.02

说明：中枢 ZD 值，非构成中枢的 K 线该值返回 0。

3.3.20 zsgg

函数名：zsgg

参数：无

依赖：zs

since：1.02

说明：中枢 GG 值，非构成中枢的 K 线该值返回 0。

3.3.21 zsdd

函数名：zsdd

参数：无

依赖：zs

since：1.02

说明：中枢 DD 值，非构成中枢的 K 线该值返回 0。

3.3.22 zszz

函数名：zszz

参数：无

依赖：zs

since：1.02

说明：中枢中轴值，中枢震荡监视器，非构成中枢的 K 线该值返回 0。

3.3.23 zskslast

函数名：zskslast

参数：无

依赖：zs

since：1.02

说明：距离上一个中枢开始的时间周期，如果中枢不存在返回-1

3.3.24 zsjslast

函数名：zsjslast

参数：无

依赖：zs

since：1.02

说明：距离上一个中枢结束的时间周期，如果中枢不存在返回-1

3.4 函数使用注意事项

- 1) 本插件所有函数，兼容大智慧和飞狐。
- 2) 函数名大写的函数为可单独调用的函数，函数名小写的函数为不可单独调用的函数(单独调用得不到数据)。

例：

压力线:"chanlun@JXYL",colorgreen,linethick1; {JXYL 可直接单独调用}

kx:="chanlun@kxian";

kxno:"chanlun@kxno", linethick0, coloryellow; {kxno 必须在 kxian 之后调用才有效果}

- 3) 函数如无特别说明，均不需要参数。具体参考详细函数说明。
- 4) 需注意小写函数之间依赖关系。kxian 是本插件所有的小写函数都需要依赖的函数，使用时 kxian 要放在所有其他的小写函数之前调用。具体参考详细函数说明。

如：(->表示依赖于)

bilast->bi

zs-gg->zs

zs->duan->bi->kxian

- 5) 走势中枢(zs)表示的是在看到的 K 线图中以由笔构成的中枢。所有能看到的中枢都是已完成的走势的中枢，当下需要自己去判断。

6) 和悟多插件兼容的三个函数为: kxian、bi、duan, 均以 1 -1 表示 顶底 (本插件中其他值的含义请参考详细函数说明)

7) 关于笔段(bi、duan), 以下假设 1F 图为 f(0), 精准的走势需要全部从 1F 线开始推导, 走势的级别是客观存在的, 是与图形的时间周期是没有任何关系的, 但是大部分筒子用的软件都没有全部也不可能有全部的 1FK 线数据的, 所以这里利用图形的时间周期的笔段来表达这么一种近似关系, 偶再罗嗦下走势的级别是客观存在的, 是与图形的时间周期是没有任何关系的。这里这种近似关系本质上相当于放大镜, 不影响大走势的分析, 而处理起来比从 1F 推导简单又容易的多。

1F 图: bi=笔 duan=段 zs=类中枢

5F 图: bi=段 duan=1F 走势 zs=1F 走势中枢

30F 图: bi=1F 走势 duan=5F 走势 zs=5F 走势中枢

日线图: bi=5F 走势 duan=30F 走势 zs=30F 走势中枢

这个不一定准确, 有时候可能 15F 图和 60F 图更能反应走势。