

 **TRADESIGNAL**
USER MANUAL



TABLE OF CONTENTS

GETTING STARTED

User Interface in a Nutshell	8
Important Features Tour	11
Using the Help Button	20

USER INTERFACE

Workspaces	22
Command Line	28
Toolbar	32
Quick Access Toolbar	34
Toolbox	35
Keyboard Shortcuts	39

FINDING AND HANDLING SECURITIES

Symbol Search	44
Symbol Lists	47
Scanner	53
Watchlist	59
Continuous Contracts	65
Market Depth	71
Filter	74

CHART TYPES

Bar Chart	80
Candlestick Chart	82
Line Chart	85
Stepped Line Chart	87
Symbol Chart	88
Filled Area Chart	89
Forest Chart	90
Tutorial Indicator as Forest	91
Linked Forest Chart	94
Candle Volume Chart	95
Equi Volume Chart	97

Kagi Chart	100
Renko Chart	102
Three Line Break Chart	104
Heikin Ashi Chart	106
Point&Figure Chart	109
Market Profile	113

DRAWING TOOLS

Text	118
Trend Line	119
Trend Channel	122
Stop Line	125
Andrew's Pitchfork	127
Linear Regression Channel	130
Gann Line	133
Arrow	135
Guide Line	137
Ellipse	139
Rectangle	140
Circle	141
Arc	142
Fibonacci Retracements	143
Fibonacci Extensions	147
Fibonacci Time Projection	150
Fibonacci Time Zones	152
Fibonacci Arcs	154
Fibonacci Fans	156

CHARTING

Chart	158
Chart Properties	168
Chart Tools	174
Timespans and Periods	179
Using Templates	182
Styles	184
Forward Curve	186
Percent Performance	188
Seasonal Chart	190

STATISTICS

Total Commission and Slippage	194
Max. Drawdown	195
Froehlich Factor	196
Sharpe Ratio	197
Percent Profitable	198
Average Trade Net Profit	199

TRADING SYSTEMS AND ORDER ROUTING

Using Trading Strategies	201
Trading Strategy Wizard	202
Money Management	208
Portfolio	215
Position Manager	222
Optimizer	226
Performance Report	233
Position Sizing - Various Methods	237
Automatic Order Routing	242

MISCELLANEOUS FEATURES

Tradesignal Options	250
Advanced Installation Options	257
Alerts	258
Package and File Management	262
Print	266
Save & Send	272
Web Browser	277
HTML Editor	278
News	285

WORKING WITH EQUILLA

Equilla Basics	290
Equilla Editor	292
Equilla Debugger	294
Equilla Program Structure and Syntax	301
Equilla Functions	308
Equilla Operators	314
Equilla Loops	320
Equilla Conditions	323

Equilla Quotations and Time Periods	326
Equilla Graphs, Outputs and Alerts	337
Equilla Arrays	351
Equilla Vectors	357
Extending Equilla	367

FUNCTIONS, INDICATORS AND STRATEGIES

Introduction to Functions, Indicators and Strategies	368
Using Indicators	372
Writing Indicators	378
Using Strategies	381
Writing Strategies	386

EXTENDING EQUILLA

C-API Extensions	394
COM-API Extensions	403
How to write a COM Extension	406
How to deploy a COM Extension	411
IEquillaExtension Interface	412
IEquillaHost Interface	413
ISupportSessions Interface	415
IEquillaExtensionSession Interface	416
EquillaConstants Class	417
Extension Signals and Sinks	418
Extension Supported Parameter Types	423

DATA MANAGEMENT

Data Feeds	432
Sessions, Holidays and Properties	433
Importing External Data	436
Tradesignal data in Excel	444
Exporting Price Data	446
XML Data Servers	447
Price Editor	449

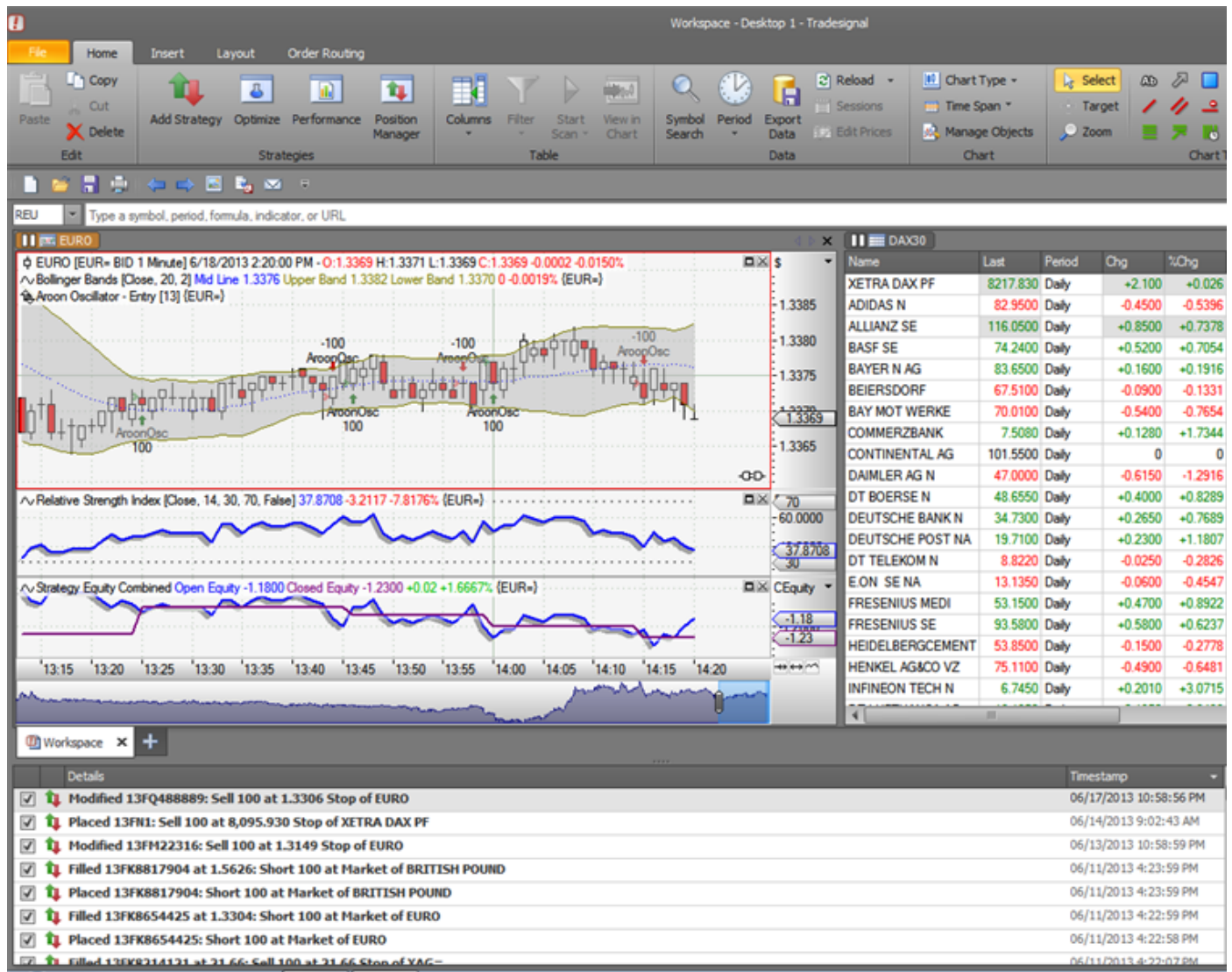
MISCELLANEOUS

Reuters Data	452
Bloomberg Data	454

GETTING STARTED

USER INTERFACE IN A NUTSHELL

When you start Tradesignal, the main user interface will be shown:



The Tradesignal User Interface

- At the very top area of the window, there is the title bar. It contains information about the current window and the Quick Access Toolbar.
- Below the title bar, there is the Toolbar (or Ribbon bar). It offers access to all basic functionality (Edit, Insert, Search...). The Toolbar is tabbed with different toolbars for different groups of functionality. New tab pages may appear depending on the document currently selected in a workspace.
- To the right of the toolbar tabs, you can see the status of the connection to the data provider. A green icon shows that the connection is available, and a red icon signals a failed connection. To reconnect a failed connection, double-click the icon to open the Primary Data Feed (Online Data) Settings in the Tradesignal Options.
To the right of the status icon, the time (depending on the time zone) is displayed. When scheduled print jobs are active, an

icon is displayed to the left of the status icon.

When you move the mouse pointer onto the status icon, a status pop-up opens. Here you can find information about

- the used data provider
- the current connection status
- the connected server (depending on the used data provider)
- possibly active print jobs
- the set timezone
- On the right, you can find the Toolbox. It offers access to symbols, properties, indicators, strategies, functions, templates, (saved) workspaces and help. Above the button area, the details are displayed, like symbol lists or the property manager for the selected elements.
- In the middle, you find the open Workspaces. This is where document types like charts, portfolios, watchlists, scanners, indicators, strategies, and other information are displayed. The tabs below the currently visible workspace allow selection of which workspace to show.
- Below the workspaces, there is the output window which contains tabs for the alerts generated by strategies, tools and indicators.

USING MOUSE AND KEYBOARD

MOUSE HANDLING

The user interface is operated with the mouse.

- Elements and buttons are activated or selected by clicking on them with the left mouse button (default). Typical activities are clicking on buttons to start activities, clicking on menu entries, or clicking on a drawing object to select it. When chart objects are selected, their properties become available in the property inspector in the toolbox.
- To open the context menu of an element, click on it with the right mouse button (also called *right-click*). The context menu opens, and you can select entries in the menu by clicking the default (left) mouse button. Context menus are available almost everywhere, e.g. for documents, indicators etc.
- Some elements or menus open with a double click. This means clicking two times in fast succession with the left mouse button. (If you click too slowly, you may simply open and close an element. In this case, try double-clicking again.) Typical double-click activities are opening symbol lists (instead of clicking on the plus symbol), or opening a symbol in a chart (instead of selecting **Open in Chart** from the context menu).
- You can move elements around with the mouse. To do so, you need to *drag&drop* the elements. Click on the element with the left mouse button and keep the button pressed while moving the pointer to a new position (drag). Once there, release the mouse button (drop). Typical drag&drop activities in Tradesignal are applying indicators to symbol charts by dragging them into the chart, or drawing lines, squares etc. with the chart tools.

KEYBOARD HANDLING

Many elements can be opened/closed, or actions can be started, by pressing a certain keyboard shortcut. These shortcuts are notated in a special way, for example:

- **Ctrl + S** means "press the **Ctrl** key and the **S** key at the same time".

- **Ctrl + Shift + F4** means "press the **Ctrl** key, the **Shift** key and the **F4** function key all at the same time". (Function keys are usually in the top area of a keyboard.)

You can find a comprehensive list of all shortcuts in the chapter Keyboard Shortcuts.

HIDING USER INTERFACE ELEMENTS

You can hide the toolbar, the toolbox and the alert area.

- To hide the Toolbar, click on the icon with the vertical arrow at the top right in the header bar. To display the hidden Toolbar temporarily, left-click onto one of the toolbar tabs that remain visible.
- To hide the Toolbox, click on the icon with the small right-pointing arrow in the toolbox title. The toolbox is hidden to the very right, under a long button labeled with the currently selected toolbox panel. Click on this button to temporarily open the toolbox, or click the left-pointing arrow button at the top of the bar to show the full toolbox once more.
- To hide the alert area (which is situated in the lower area of the workspace), click on the button to the bottom-right of the pane, below the tabs. When hidden the alert area will be replaced with a small info icon next to the workspace tabs, click this to restore the alert pane. The alert pane can also be maximized using the button above the collapse button. You can find more information on alerts in the chapter Alerts.

IMPORTANT FEATURES TOUR

In this article you can find an overview of the most important Tradesignal functions. For detailed information, please follow the references to the respective articles.

CLASSIC CHART ANALYSIS

Tradesignal makes it easy for you to create charts for one or more symbols, indicators and strategies. The design possibilities are vast. In the following, some important features will be given.

You can find an introduction to the user interface in the chapter User Interface in a Nutshell.

CREATING A CHART

To open a new empty chart, click on the **Chart** button in the Toolbar. You can then add symbols to this chart, e.g. by dragging symbols from the Toolbox into it.

You can also quickly open a chart for a symbol directly from the toolbar:

1. Click on the **Symbol Lists** button in the Toolbox.
2. Open a symbol list, e.g. the Nasdaq indices, and look for the ".NDX".
3. Double-click on the entry. Alternatively, right-click on it and select **Open in Chart** in the context menu.

You can also enter the symbol shortcut in the Command Line.

1. Enter the shortcut ".NDX" in the command line.
2. Select **New Chart** from the drop-down menu.

The data is displayed as a candlestick chart. You can change this default chart setting in the Tradesignal Options, see the chapter Set New Default Chart Style. The line and bar chart types are also available as default.

In general, many more chart types are available to you, like Forest, Equivolume, Kagi or Point&Figure. You can find a complete list of all available chart types in Chart.

On the top left in the (sub-)chart, you can find the chart legend. Here, information about the symbols, indicators, strategies, periods etc. is available. The chart legend offers a context menu in which you can find various functions, depending on the entry type.

CHANGING THE PERIOD AND TIME SPAN



Chart

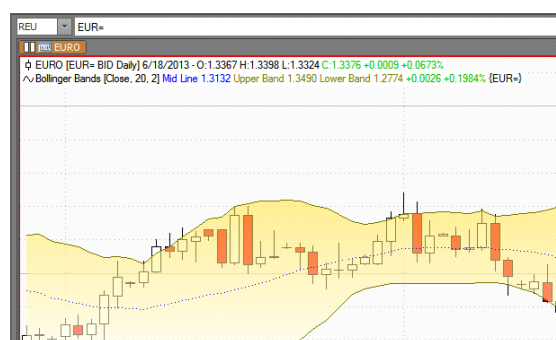


Chart Legend

There are various ways to change the displayed data. In the toolbar, the buttons **Period** and **Time Span** are available. Click on the small arrows to open the respective button menus.

- Keep the **Period** at **Daily**. With this setting, one bar (candlestick) is drawn per day.
- For the **Time Span**, select **Current Year**, for example.

You can also change the displayed time span and prices by directly manipulating the axes with the mouse. To do so, click on the time axis with the right mouse button and keep it pressed while moving the mouse left and right. You can see that the axis is stretched and shrunk. The price axis can be changed accordingly. You can find more information in the chapter Stretch/shrink Axes.



Editing the Period

ADDING DRAWING ELEMENTS

In the chart tools you can find simple drawing elements like lines, circles and rectangles, but also tools with a calculation function, like the Fibonacci Retracements or Andrew's Pitchfork. You can find more information in the chapter Chart Tools.

1. Click on the symbol for the trend line in the toolbar.
2. With the mouse button pressed, draw a trend line between two points in your current chart. A line is drawn.
3. Now click on the symbol for the linear regression channel in the toolbar.
4. With the mouse button pressed, draw a linear regression channel from a low to a high in the chart. You can see that the midline and a regression channel are calculated and drawn; they change angle and width depending on the bars (candlesticks) that are included from the start to the end point.



Trend Line and Regression Channel in the Chart

Some drawing elements can be combined with the alert function. You can find more information in the chapter Alerts.

CHANGING THE CHART DESIGN

You can edit the details of the chart design, for example by changing the chart properties in the toolbox.

However, you can also use one of the available chart styles to change the design extensively.

1. Click on the small arrow on the **Styles** button in the toolbar to open the button menu.
2. Select a style from the list, for example **Blue**. If you have drawing elements in the chart, you can see that changing the style also



Edited Chart Style

results in color changes for the drawn elements.

You can find more information in the chapter Chart Properties.

DISPLAYING SEVERAL SYMBOLS IN THE SAME WORKSPACE

You can display more than one symbol in the same workspace, e.g. as subchart. For that, it is important to have a look at the settings for subcharts in the Tradesignal Options.

1. Open the file menu (above the toolbar in the Tradesignal window) and select the entry **Options**.
2. On the left, click on the entry **Advanced** and then check whether the setting **Instruments > Open symbols in a new subchart** is selected.

This entry has the following effect:

- If no subcharts should be opened: If a chart with a symbol is displayed and you add a symbol by selecting **Add Symbol** in the context menu of that symbol, then the symbol is displayed in the same chart as the first symbol. This can make it hard to distinguish between the two quotations.
- If subcharts should be opened: The added symbol is displayed as a subchart below the first symbol.

Subcharts share the same time and price axes. This has the advantage (or disadvantage) that all changes of e.g. period and time span apply to all subcharts.

Tip: You can edit the period for a single symbol in the chart legend context menu.

You can also display several charts in one workspace. For this, simply add the symbol by selecting **Open in Chart** in the symbol context menu. This way, the chart opens with its own axes.

You can find more information in the chapter Charts.

KEEPING THE SYMBOL OVERVIEW

SYMBOLS AND SYMBOL LISTS



Three Charts as Subcharts



Three Charts as Charts

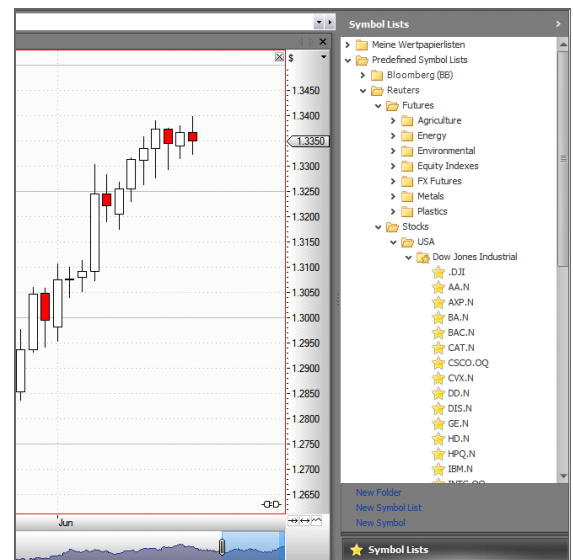
You can see symbols and symbol lists by clicking on the button **Symbol Lists** in the toolbox.

Symbol lists are sorted in groups. These are labeled with a folder symbol. Click on the plus symbol in front of it to display the symbol lists.

Symbol lists are labeled with a document symbol with a star. Symbols are labeled with a star.

The context menu offers a lot of functions for both. For example, you can open symbols in charts or in a market depth document. Symbol lists can be opened in a Scanner, Watchlist or a Portfolio.

You can find more information in the chapter Symbol Lists and the chapter Symbol Search.



Symbols and Symbol Lists

ADDING SYMBOLS TO LISTS

Single symbols are always managed in symbol lists.

1. To add a new symbol, open the context menu of a symbol list and select **New Symbol**. The New Symbol wizard opens.
2. You can choose between three options. Select **A standard symbol (from the data provider)**.
3. The Symbol Search opens. Search the symbol you want to add.

You can also add symbols to symbol lists by dragging symbols into them from other lists.

You can find more information in the chapter Add a Symbol to a Symbol List.

CREATING A NEW SYMBOL LIST

You can create your own symbols lists, for example to have your favorites in one list.

1. In the toolbox, click the **Symbol Lists** button and then on **New Symbol List** in the *Related Task* area. The Symbol List wizard opens. Select the option **User-defined Symbol List**.
2. The second window of the Wizard opens. Enter a name for the symbol list.
3. Click on **Add** to start the Symbol Search. Find the symbols, select them and **Add** them to the list.
4. Click on **Done**. The new symbol lists appears in the toolbox.

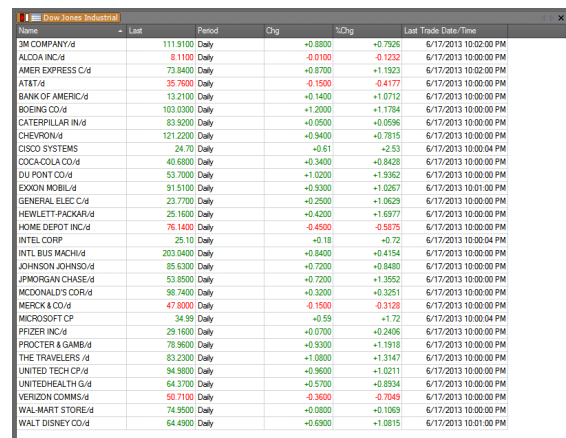
You can find more information in the chapter Symbol Lists.

USING A WATCHLIST

The Watchlist is optimized for monitoring symbol data in realtime. In combination with indicators (see below), alerts can be issued when certain data is "interesting" as defined by your indicators.

The easiest way is to open an existing symbol list in a Watchlist.

1. Right-click on a symbol list in the toolbox and select **Open in Watchlist** from the context menu. The Wizard opens.
2. You can add one or more indicators, for example the "Relative Strength Index".
3. In the next step, you can also add one or more strategies.
4. When finished selecting, click on **Done**.



Name	Last	Period	Chg	%Chg	Last Trade Date/Time	RSI
3M COMPANY/i	111.9100	Daily	-0.8800	-0.7926	6/17/2013 10:02:00 PM	46.85
ALCOA INC/i	8.1100	Daily	-0.0100	-1.2332	6/17/2013 10:00:00 PM	29.6914
AMER EXPRESS C/i	73.8400	Daily	+0.8700	+1.1923	6/17/2013 10:02:00 PM	47.2389
AT&T/i	35.7600	Daily	-0.1500	-0.4177	6/17/2013 10:00:00 PM	41.3170
BANK OF AMERICA/i	13.2100	Daily	+0.1400	+1.0712	6/17/2013 10:00:00 PM	50.2383
BOEING CO/i	103.0300	Daily	+1.2000	+1.1784	6/17/2013 10:00:00 PM	60.1619
CATERPILLAR IN/i	83.9200	Daily	+0.0500	+0.0596	6/17/2013 10:00:00 PM	33.8452
CHEVRON/i	121.2200	Daily	+0.9400	+0.7815	6/17/2013 10:00:00 PM	67.48
CISCO SYSTEMS	24.70	Daily	-0.61	-2.53	6/17/2013 10:00:04 PM	38.9035
COCA-COLA CO/i	40.6800	Daily	+0.3400	+0.8428	6/17/2013 10:00:00 PM	40.5375
DU PONT CO/i	53.7000	Daily	+1.0200	+1.9362	6/17/2013 10:00:00 PM	47.5904
EXXON MOBIL/i	91.5100	Daily	-0.9300	-1.0267	6/17/2013 10:01:00 PM	43.1493
GENERAL ELEC C/i	23.7700	Daily	+0.2500	+1.0629	6/17/2013 10:00:00 PM	48.6047
HEWLETT-PACKARD/i	25.1600	Daily	+0.4200	+1.6977	6/17/2013 10:00:00 PM	52.7967
HOME DEPOT INC/i	76.1400	Daily	-0.4500	-0.5875	6/17/2013 10:00:00 PM	39.2254
INTEL CORP	25.10	Daily	+0.18	+0.72	6/17/2013 10:00:04 PM	63.00
INTL BUS MACH/i	203.0400	Daily	+0.8400	+0.4154	6/17/2013 10:00:00 PM	44.1507
JOHNSON JOHNISO/i	85.6300	Daily	+0.7200	+0.8480	6/17/2013 10:00:00 PM	43.7475
JPMORGAN CHASE/i	53.8500	Daily	+0.7200	+1.3552	6/17/2013 10:00:00 PM	52.5934
MCDONALD'S COR/i	98.7400	Daily	+0.3200	+0.3251	6/17/2013 10:00:00 PM	42.1496
MERCK & CO/i	47.8000	Daily	-0.1500	-0.3128	6/17/2013 10:00:00 PM	42.9591
MICROSOFT CP	34.99	Daily	+0.59	+1.72	6/17/2013 10:00:04 PM	57.58
PFIZER INC/i	29.1600	Daily	+0.0700	+0.2406	6/17/2013 10:00:00 PM	50.7101
PROCTER & GAMB/i	78.9600	Daily	+0.9300	+1.1918	6/17/2013 10:00:00 PM	51.2679
THE TRAVELERS /i	83.2300	Daily	+1.0800	+1.3147	6/17/2013 10:00:00 PM	49.3769
UNITED TECH CP/i	94.9800	Daily	+0.9600	+1.0211	6/17/2013 10:00:00 PM	48.6047
UNITEDHEALTH G/i	64.3700	Daily	+0.5700	+0.8934	6/17/2013 10:00:00 PM	60.5953
VERIZON COMMS/i	50.7100	Daily	-0.3600	-0.7049	6/17/2013 10:00:00 PM	43.1493
WAL-MART STORE/i	74.9500	Daily	+0.0800	+0.1069	6/17/2013 10:00:00 PM	41.6256
WALT DISNEY CO/i	64.4900	Daily	-0.6900	-1.0815	6/17/2013 10:01:00 PM	49.2628

Watchlist

Similarly you can open single symbols in the Watchlist.

To open an empty Watchlist, click on the **Watchlist** button in the toolbar.

You can sort the Watchlist columns by clicking on the respective column header. A small triangle appears (pointing up or down, depending on the sorting direction). To change the direction, click on the column header again.

Symbols can be copied from other tables into the Watchlist by using Drag & Drop, e.g. from the Scanner to the Watchlist. In a similar way, you can drag symbols from the Watchlist into a Scanner or Portfolio.

1. Click into a table cell of the symbol you want to copy. A small plus sign and a rectangle appear under the mouse cursor.
2. Drag the cursor into the other table. The symbol is copied into the table.

You can find more information in the chapter Watchlist.

SCANNING SYMBOLS

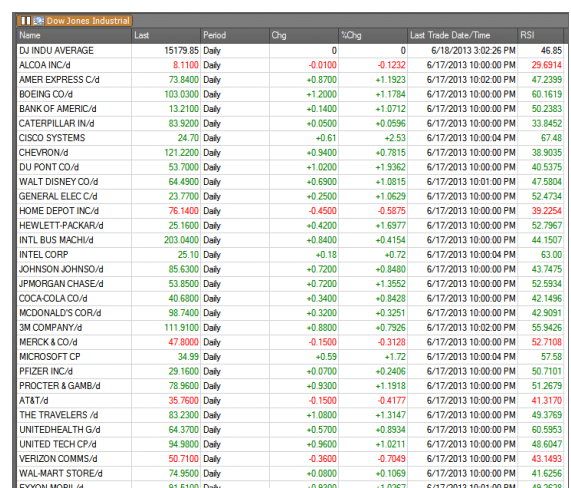
The Scanner helps you search the large number of symbols traded at international markets for profitable trading candidates.

In contrast to the Watchlist, the Scanner offers no realtime information. It is therefore more useful for scanning large symbol lists for indicator or price patterns, for example overnight.

Similar to the Watchlist, you can quickly fill the Scanner by calling up the context menu of a symbol or symbol list and selecting **Open in Market Scanner**.

As for a Watchlist, you can

- open an empty Scanner via the toolbar
- directly add indicators and strategies when creating a Scanner
- copy symbols from the Scanner into other tables by using Drag&Drop, or filling symbols from other tables into the Scanner
- sort the columns in the Scanner by clicking on the respective column headers.



Name	Last	Period	Chg	%Chg	Last Trade Date/Time	RSI
DJ INDU AVERAGE	15179.85	Daily	0	0	6/18/2013 3:02:26 PM	46.85
ALCOA INC/i	8.1100	Daily	-0.0100	-1.2332	6/17/2013 10:00:00 PM	29.6914
AMER EXPRESS C/i	73.8400	Daily	+0.8700	+1.1923	6/17/2013 10:02:00 PM	47.2389
AT&T/i	35.7600	Daily	-0.1500	-0.4177	6/17/2013 10:00:00 PM	41.3170
BANK OF AMERICA/i	13.2100	Daily	+0.1400	+1.0712	6/17/2013 10:00:00 PM	50.2383
CATERPILLAR IN/i	83.9200	Daily	+0.0500	+0.0596	6/17/2013 10:00:00 PM	33.8452
CHEVRON/i	121.2200	Daily	+0.9400	+0.7815	6/17/2013 10:00:00 PM	67.48
CISCO SYSTEMS	24.70	Daily	-0.61	-2.53	6/17/2013 10:00:04 PM	38.9035
COCA-COLA CO/i	40.6800	Daily	+0.3400	+0.8428	6/17/2013 10:00:00 PM	40.5375
DU PONT CO/i	53.7000	Daily	+1.0200	+1.9362	6/17/2013 10:00:00 PM	47.5904
EXXON MOBIL/i	91.5100	Daily	-0.9300	-1.0267	6/17/2013 10:01:00 PM	43.1493
GENERAL ELEC C/i	23.7700	Daily	+0.2500	+1.0629	6/17/2013 10:00:00 PM	48.6047
HEWLETT-PACKARD/i	25.1600	Daily	+0.4200	+1.6977	6/17/2013 10:00:00 PM	52.7967
HOME DEPOT INC/i	76.1400	Daily	-0.4500	-0.5875	6/17/2013 10:00:00 PM	39.2254
INTEL CORP	25.10	Daily	+0.18	+0.72	6/17/2013 10:00:04 PM	63.00
INTL BUS MACH/i	203.0400	Daily	+0.8400	+0.4154	6/17/2013 10:00:00 PM	44.1507
JOHNSON JOHNISO/i	85.6300	Daily	+0.7200	+0.8480	6/17/2013 10:00:00 PM	43.7475
JPMORGAN CHASE/i	53.8500	Daily	+0.7200	+1.3552	6/17/2013 10:00:00 PM	52.5934
MCDONALD'S COR/i	98.7400	Daily	+0.3200	+0.3251	6/17/2013 10:00:00 PM	42.1496
MERCK & CO/i	47.8000	Daily	-0.1500	-0.3128	6/17/2013 10:00:00 PM	42.9591
MICROSOFT CP	34.99	Daily	+0.59	+1.72	6/17/2013 10:00:04 PM	57.58
PFIZER INC/i	29.1600	Daily	+0.0700	+0.2406	6/17/2013 10:00:00 PM	50.7101
PROCTER & GAMB/i	78.9600	Daily	+0.9300	+1.1918	6/17/2013 10:00:00 PM	51.2679
THE TRAVELERS /i	83.2300	Daily	+1.0800	+1.3147	6/17/2013 10:00:00 PM	49.3769
UNITED TECH CP/i	94.9800	Daily	+0.9600	+1.0211	6/17/2013 10:00:00 PM	48.6047
UNITEDHEALTH G/i	64.3700	Daily	+0.5700	+0.8934	6/17/2013 10:00:00 PM	60.5953
VERIZON COMMS/i	50.7100	Daily	-0.3600	-0.7049	6/17/2013 10:00:00 PM	43.1493
WAL-MART STORE/i	74.9500	Daily	+0.0800	+0.1069	6/17/2013 10:00:00 PM	41.6256
WALT DISNEY CO/i	64.4900	Daily	-0.6900	-1.0815	6/17/2013 10:01:00 PM	49.2628

Symbols in the Scanner

You can find more information in the chapter Scanner.

INDICATORS AND STRATEGIES

Indicators and strategies are the main themes in analyzing symbol quotations.

Alongside classic drawing tools like trend lines or channels, indicators help to classify trends or offer clear trading signals. In Tradesignal, you can use indicators with Charts, Scanners, Watchlists and more. Indicators can be combined, derived from each other, or calculated for more than one symbol.

Strategies combine indicators with entry and exit signals. This allows you to trade mechanical systems.

You can find more information in the chapter Introduction to Functions, Indicators and Strategies.

APPLYING AN INDICATOR TO A CHART

1. Open a symbol in a chart.
2. Click on the **Indicators** button in the toolbox.
3. Drag the indicator "Moving Average Exponential" onto the chart legend of the symbol. Alternatively, you can also select **Apply** in the context menu of the indicator.

In a similar way you can apply indicators to a Watchlist, Scanner and Portfolio.

You can find more information in the chapter Using Indicators.

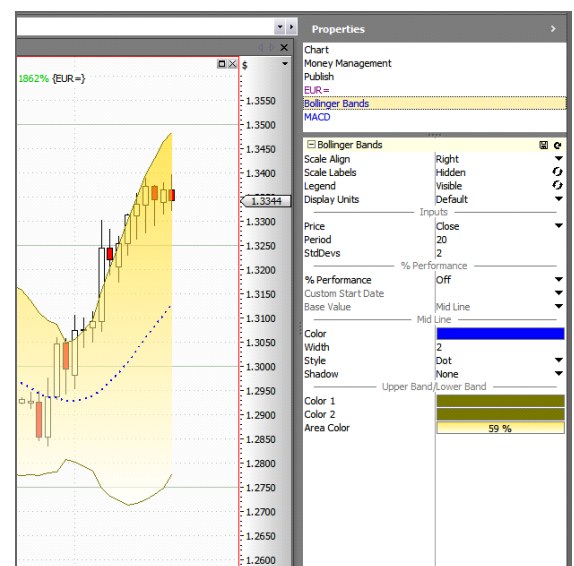
EDITING THE INDICATOR PROPERTIES

You can edit the properties of indicators.

- Right-click the indicator entry in the chart legend and select **Properties** from the context menu. The indicator properties are displayed in the toolbox.
- Alternatively, click on the **Properties** button in the toolbox and select the indicator in the list of elements available in the chart, on top of the properties display. (For example, for a chart that may be "Chart", "Money Management", the symbol shortcut, indicators and strategies.)

For the "MACD" indicator in our example, you can set the **PeriodSlow** to "38" by clicking into the field and entering a number. You can also click on the spin buttons to increase or decrease the number.

APPLYING AN INDICATOR TO AN INDICATOR



Indicator Properties in the Toolbar

You can apply a second indicator to the indicator used above, "Moving Average Exponential", for example via the command line.

1. Click on the "Moving Average Double Crossover" entry in the chart legend to select the indicator.
2. Enter the symbol shortcut **MACD** in the command line.
3. Select **Add Indicator** in the drop-down menu.

The "MACD" indicator is now applied to the "Moving Average Exponential".



Applying an Indicator to an Indicator

APPLYING STRATEGIES TO A CHART

Strategies are an important part of trading systems, since they combine indicators with entry and exit signals. This allows you to trade mechanical systems.

Strategies are applied similarly to indicators. You can apply strategies to Charts, Watchlists, Scanners and Portfolios.

Here is an example that you can also use for the optimization below.

1. Open a symbol in a chart, for example a symbol of the Dow Jones group.
2. Click on the **Strategies** button in the toolbar.
3. Select the "Moving Average Double Crossover" strategy and drag it into the chart.
4. Click on the **Properties** button and select the strategy in the list of elements on top.
5. Set **PeriodFast** to "5" days and **PeriodSlow** to "10" days.

Now the strategy is set up.



Applying Strategies

Alternatively, you can add trading systems by clicking on the **Add Strategy** button in the *Trading Systems* group in the toolbar. A window opens in which you can find strategies sorted by subject, e.g. strategies for stops or position sizing.

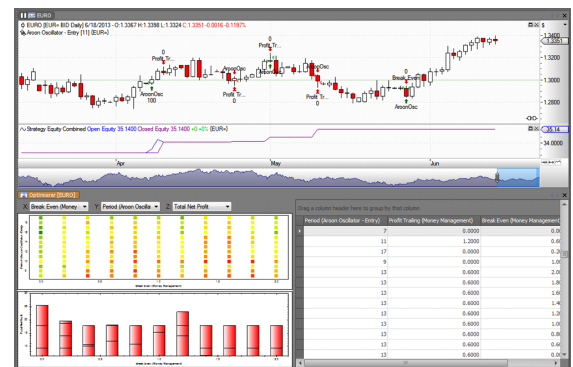
You can find more information in the chapter Using Strategies.

OPTIMIZING TRADING SYSTEMS

Trading systems are often based on mathematical or statistical calculations and include many variables. The Optimizer helps you to find good and stable parameter sets for your trading system.

For the following example, please use the system you created in the previous section.

1. Click on the **Optimize** button in the toolbar.
2. Select the method **Brute Force**.
3. In the right field, select the parameter **PeriodFast**.
4. Remove it by clicking on the << button in the middle of the window.
5. Select **PeriodSlow** in the field on the right and enter the start value "10" and the end value "50".
6. Click on **Next**. Click on **Next again** (so that all available data is used for optimization) and finally click on **Optimize** in the last window.



Results of the Optimizer

The optimization starts. This may take some time, therefore a button **Pause** is available in the *Optimizer* group in the toolbar. Click on it to halt the optimization and to display the intermediate results.

When the optimization is finished, the results are displayed, in this case for optimizing the **Moving Average Double Crossover** strategy. During the optimization, Tradesignal has calculated and checked many possible parameter combinations, and the result shows that certain settings for **PeriodSlow** would have been more profitable than others.

To apply the result of the optimization, for example the number for **PeriodSlow** with the highest profit, simply double-click on the row with the highest profit and confirm the warning with **Yes**.

Please remember that the optimization is always a backtest, e.g. working with the data of past trading periods. Therefore it is recommended to check the optimization results against current developments from time to time.

You can find more information in the chapter Optimizer.

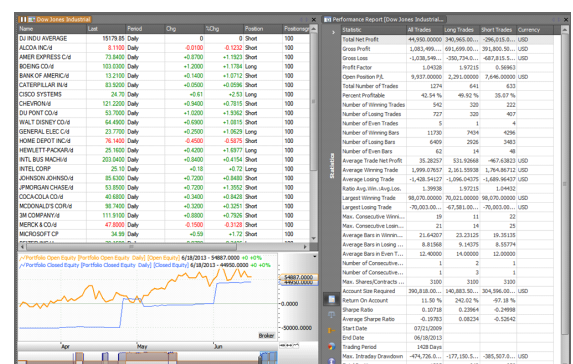
PORTFOLIO TRADING

Trading with a trading system of one or more symbols, or even several trading systems, is done most effectively by using the Portfolio function. With it, you can trade via a central account that includes settings for money and risk management. The Portfolio also allows you creating a global Performance Report.

In a similar way to the Scanner and Watchlist, you can open an empty Portfolio from the toolbar, or you can open symbols and symbol lists in a Portfolio by selecting **Open in Portfolio** from the context menus.

As for a Watchlist, you can

- directly add indicators and strategies when creating a Portfolio



Portfolio Trading

- copy symbols from the Portfolio into other tables by using Drag & Drop, or filling symbols from other tables into the Portfolio
- sort the columns in the Portfolio by clicking on the respective column headers.

You can find more information in the chapter Portfolio.

POSITION MANAGER

When trading a mechanical system, the Position Manager offers a complete overview over your current trading system activities. It displays open, filled, and canceled orders and your currently held positions.

You can open the Position Manager via the button **Position Manager** in the toolbar. However, there are only entries in the Position Manager if a number of conditions have been met.

You can find more information in the chapter Position Manager.

The Position Manager is closely correlated with automatic order routing, see next paragraph.



Position Manager

AUTOMATIC ORDER ROUTING

The automatic order routing function in Tradesignal allows you trading with a more or less mechanical trading system connected to your broker. You can decide if the orders are to be routed automatically or only with a manual confirmation.

If you are using broker software without interfaces to Tradesignal, you can set up the order routing so that the trading activity is only displayed in the Position Manager. You can then enter the orders into the broker's software manually.

You can find more information in the chapter Automatic Order Routing.

USING THE HELP BUTTON

With the help button **File** menu you can:

- call up the help topics of Tradesignal
- write an e-mail to support with necessary system information
- open *Tradesignal online* discussion forums in the internal browser
- show the application log file
- search for application updates
- get information about your current Tradesignal version

HELP TOPICS

Show the online version of the help topics as found in the help toolbox. The help topics give you an overview of Tradesignal help. You can either manually search the tree structure for an article or you can use the input field to search for a specific keyword. In this case, a flat list of articles with those keywords is displayed. If you want to go back to the overview, click on the link *Table of Contents*.

SEND SUPPORT INFORMATION

This function collects some data about your Tradesignal system, puts it into a ZIP file, so that it can be attached to your email to send to support. This data collection will help the support team solve your problem.

CHECK FOR UPDATES

Here you can check for the latest update of Tradesignal. This options is not available in all versions of Tradesignal.

ABOUT TRADESIGNAL

Shows you information about your Tradesignal installation.

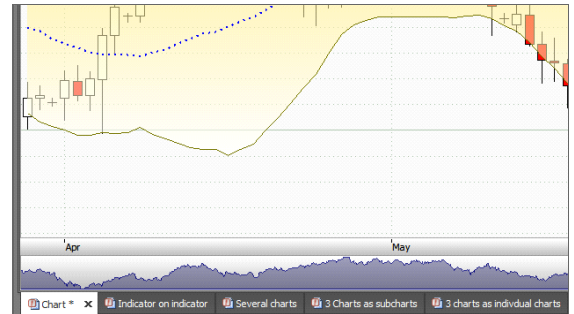
USER INTERFACE

WORKSPACES

The workspace is the area of a software desktop in which documents are open and editable. Tradesignal workspaces provide many special functions to make using the software easier for you.

You can use workspaces for the following:

- Editing and saving several charts at the same time
- Arranging charts in groups
- Saving different documents like charts, scanners, portfolios and editors as one workspace
- Working with several desktops and monitors at the same time



WORK WITH WORKSPACES

NEW WORKSPACE

Tradesignal always opens a new workspace if you start a new document on an empty desktop.

You can also use the following methods to set up new workspaces manually:

- Click the **New** button in the **File** menu.
- Press the key combination **Ctrl + N**.
- Click on **New** in the context menu of the workspace tab.
- Click on the plus (+) button next to the workspace tabs at the bottom of the screen.

A new workspace is opened. A new workspace tab appears on the lower left.

OPEN WORKSPACE

You can only open saved workspaces. Otherwise, you can always set up a new workspace.

Open a saved workspace:

- By opening the **Workspaces** manager in the toolbox and double-clicking a list entry or
- By clicking on **Recent** in the **File** menu and choosing a workspace from the list of last edited workspaces. If you click on the entry **More...**, a file dialog opens in which you can open, copy or move workspaces.
When you press the key combination **Ctrl + O**, a general file dialog opens in which you have access not only to workspaces but also to indicators, strategies, functions etc. In both dialogs you have the possibility to select and open more than one workspace at once.

As an additional feature, you can open workspaces via the context menu of packages. For that, the packages have to be displayed. Right-click on the package name and select either

- **Open All Workspaces** to open all workspaces in this package or

- **Open All Yellow Workspaces** (red, blue) to open the color-tagged ones only.

For color-tagging workspaces in the toolbox, see the chapter Toolbox.

RENAME WORKSPACE

To rename a workspace, open the context menu of the workspace tab and click on **Rename**. A dialog opens in which you can enter a new name for the workspace.

CLOSE WORKSPACE

To close a single workspace, choose one of the following options:

- Press the key combination **Ctrl + Shift + F4**.
- Click on **Close Workspace** in the context menu of the workspace tab or in the file menu.
- Click on the **x** button on the tab of the workspace.

To close all workspaces, click on **Close all Workspaces** in the file menu.

The last closed workspace can be quickly restored using the Restore Removed Items feature.

SAVE WORKSPACE

To save workspaces, choose one of the following entries in the file menu:

- **Save (Ctrl + S)** to save the current workspace (also available in the tab context menu).
- **Save as** to save the current workspace under another name.
- **Save all** to save all open workspaces.

You can find all saved workspaces in the **Workspaces** manager of the toolbox.

DELETE WORKSPACE

You can only delete workspaces that have been saved. Otherwise you can simply close a workspace that is no longer useful.

To delete a saved workspace, open the **Workspaces** manager in the toolbox for a list of saved workspaces. Choose **Delete** from the context menu of an entry.

NAVIGATE IN WORKSPACES

You can browse through open workspaces by pressing **Ctrl + W** for forward and **Ctrl + Q** for backward scrolling.

You can change the workspace sequence by clicking on a tab and dragging and dropping it to the left or right.

Note: If you want to see all open workspaces at a glance in multiple line, select *Multiline* in the context menu of the workspace.

DEFINE KEYBOARD SHORTCUTS

You can assign the keyboard shortcuts **Ctrl + 0** up to **9** for often used workspaces. For this, either:

- Right-click on the workspace tab or
- Right-click in the list of workspaces in the **Workspace** manager in the toolbar

and choose **Set Shortcut** to select a key command from the sub-menu.

ARRANGE WORKSPACES ON MONITORS

You can move a workspace to another monitor by clicking the workspace tab and dragging and dropping it to the target monitor.

You can also arrange the workspaces on more than one monitor by clicking on the **Layout** button in the toolbar and choosing up to ten monitors to distribute the workspaces.

ARRANGE WORKSPACES ON DESKTOPS

You can display workspaces on new desktops. Desktops are new windows that will appear in the Windows task bar. They only offer the context menus of workspace bar and chart.

To display a workspace on a different desktop, right-click the workspace tab, then choose **Change Desktop** from the sub-menu to select a desktop. Alternative you can drag-and-drop the workspace tab to the according desktop.

You can move the new desktop to other monitors by dragging and dropping.

LOCK LAYOUT AGAINST CHANGES

To prevent unwanted changes in the layout design of your workspaces, either

- click the **Layout** toolbar tab and click on **Lock**, or
- right-click the workspace tab and choose **Lock** from the context menu.

Documents in a locked workspace cannot be changed in size, form or position.

To unlock the workspace, simply click on the toolbar button a second time.

PASSWORD PROTECTION

You can restrict access to a workspace by specifying passwords for opening and editing it by selecting the **Layout** toolbar tab and clicking on **Password Protect...**

Important: *The password cannot be recovered, so if you lose it, you will not be able to edit/access your protected workspace anymore.*

ORGANIZE DOCUMENTS IN WORKSPACES

DEFAULT LAYOUTS

Tradesignal offers several default layouts for documents on a workspace. You can find them by selecting the **Layout** tab on the toolbar.

Rebalance arranges the documents as homogeneous as possible on the workspace.

Tabbed arranges all documents in a single stack. Each document is accessible via a tab.

Columns arranges all documents in a row.

Rows arranges all documents in a vertical column.

Tiled generates a checker-board pattern of the open documents. The documents are arranged as homogeneously as possible in size and position.

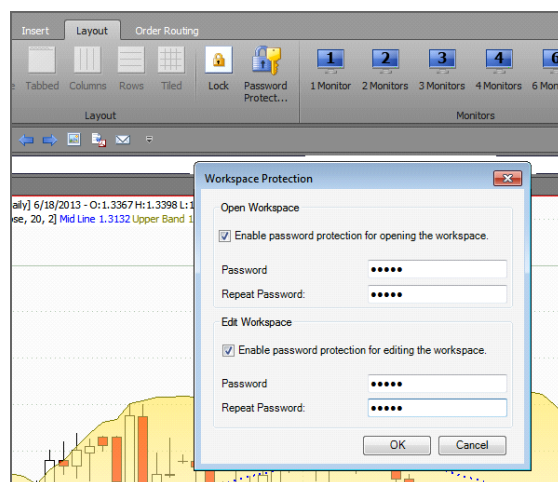
Maximize (key **F4**) shows the active document in full size in the workspace. A double-click on the document tab will also maximize the document. Repeat to return to the former arrangement.

You can change the size of a single document by moving the frame of the document to the left, right, up or down.

MOVE DOCUMENTS

You can move documents across the workspace by dragging and dropping the document tab to a new position. During dragging, the position is marked by a black, rectangular frame.

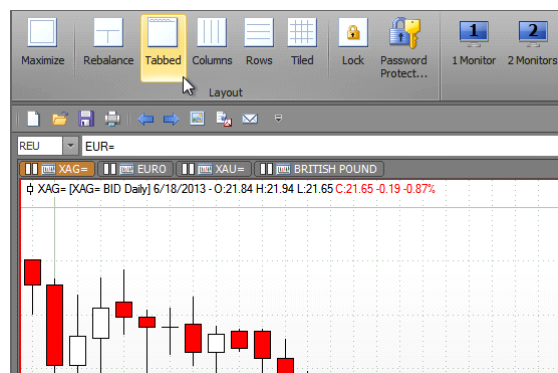
In principle, documents can be placed to the left, right, above or below another document.



Password Protection dialog



Layout "Tiled" documents



Layout "Tabs" (stacked documents)

In addition, you can place a document over another one, which results in a document stack. In this case, two tabs will be offered in the document space. The following rules apply to stacked documents:

- Click on the document tab to bring the document to the front of the stack, i.e. display.
- Drag the document tab to the left or right to change the position of the document in the stack.

QUICK NAVIGATION BETWEEN DOCUMENTS

A document is activated when you click on its tab or frame.

You can also use keyboard commands for quickly switching between documents (especially when stacked):

- **Ctrl + Tab** switches to the next document in the workspace. If documents are stacked, the next document in the stack is displayed.
- **Ctrl + Shift + Tab** switches to the previous document in the workspace. If documents are stacked, the previous document in the stack is displayed.

COPY AND PASTE DOCUMENTS

In a workspace, you can copy documents and then paste them into another or the same workspace, duplicating it.

1. Activate a document by clicking on its tab or frame.
2. In the *Edit* group in the **Home** tab of the Toolbar, click on the button **Copy** (keyboard shortcut **Ctrl + C**).
3. If necessary, switch to another workspace and click on the button **Paste** (keyboard shortcut **Ctrl + V**).

SYMBOL LINKING

You can link several documents so that in all documents the same symbol or period is displayed. The links are connected to colors, which are displayed as small filled rectangles on the top left of the document tab.

You can find the linking possibilities in the context menu of the document tab:

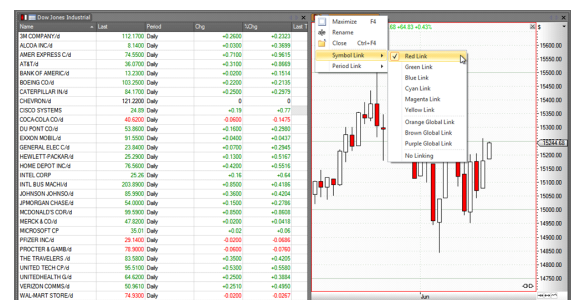
- **Symbol Linking:** When you change the symbol of a linked document, the symbol is changed in all linked documents.
- **Period Linking:** When you change the period of a linked document, the period is changed in all linked documents.

You have the choice between

- Six normal links: these links only apply within the same workspace.
- Three global links: these links apply across all workspaces.

Note that you can assign only one document and one period link per document.

- **Global component linking:** In addition to linking components within a single workspace, a component can be marked with



Link Documents

a 'global link'. The effect of this is that whenever a globally linked item is changed (either the symbol or period replaced), all other components in all open workspaces with the same global link color will also be updated.

RESTORING REMOVED ITEMS

When a workspace or item in a workspace is closed, or an item from a chart or watchlist is deleted, a yellow popup window will be displayed at the bottom of the application window with a button labelled **Unremove**. This button may be clicked to undo the last remove/delete operation. Clicking the **x** button on the popup window will hide it without performing the undo operation (a new link labelled **Unremove** will instead be displayed to the left of the clock at the top right of the application window).

To only show the unobtrusive link at the top right of the application window, uncheck the corresponding box in the options found under *File > Options > Advanced > Workspaces*.

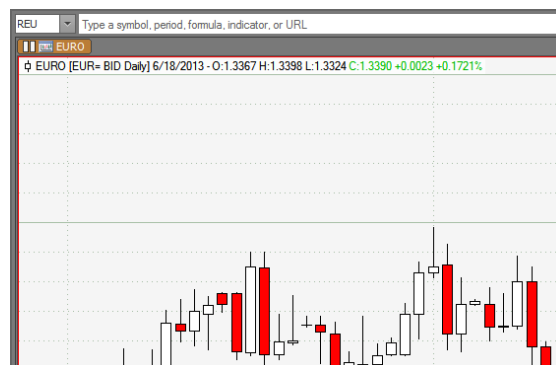
COMMAND LINE

The command line is a central control element of Tradesignal. From here, you can

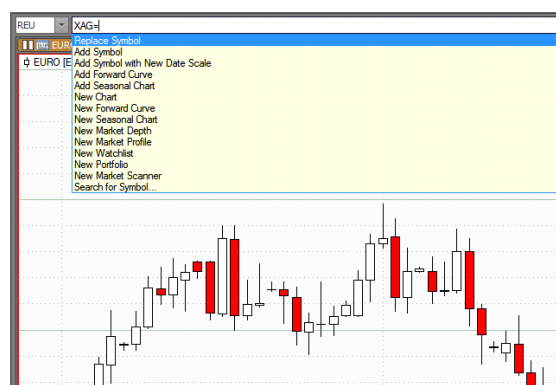
- create new documents like charts or scanners
- open Internet pages
- enter Equilla formulas
- change the trading period of charts
- enter Tradesignal-specific !Bang commands

GLOBAL COMMAND LINE FUNCTIONS

- The command line is context-sensitive, i.e. in the drop down menu of the command line, only functions available for the entered command or text are displayed. The most probable function is offered first.
- Double-click in the command line field to open the drop-down menu. This can be useful e.g. if there is a symbol shortcut in the command line and you want to call up a function for it.
- The command line "remembers" previous entries. Click the small arrow to the right of the command line field to open the list of previous entries. You can delete the command line history in the Tradesignal Options.
- On the basis of these remembered entries, the command line offers auto-completion upon entering letters. For example, if you have already entered "nsd.x" once, the next time you type "nsd" it will lead to the correct auto-completion.



Command Line



Command line with open menu

SYMBOL SHORTCUTS

With a symbol shortcut you can open a Chart, a Scanner, a Watchlist or a Portfolio. If such document types are already open, you can display the symbol in them. In case of an open chart, you can either substitute the displayed symbol or add a second time line to the existing chart.

Further commands can be entered along with the symbol shortcut:

- Periods - Enter a trading period right after the symbol shortcut. For example, ".DJI 60m" will open an hourly chart of the Dow Jones index.
- Data fields - Enter a data field name, for example "CAT NYS bid" oder "CAT NYS ask". (Calls up data from the data provider.)
- Units - Enter a unit; this is especially useful in case of commodities and goods. (Calls up data from the data provider.)

- **Currency** - Enter the currency in which the symbol should be displayed. For example, enter "IBM NYS eur" to display IBM in euros. You can find the currency shortcuts in the symbol properties in the toolbox.

SHORT CODES FOR INDICATORS

Each Tradesignal indicator, as well as many indicators from other sources offer a short code. With it you can call up the program from the command line. For example, to apply the Momentum indicator, enter "MOM" in the command line and select the offered option "Add Indicator Momentum".

Short codes can be applied to all document types that can work with indicators. To find out possible short codes, open the indicator window in the toolbox and hover over the list entries with the mouse. You can find the short code in the tool tip.

EQUILLA CODE

You can also enter Equilla Code directly in the command line (not available for trading commands like Buy or Short). This way you can easily set up new indicators or display an often-used spread between two symbols.

ENTER A FORMULA IN A CHART

You can create and apply indicator combinations by directly entering the formula in the command line. In that formula, you can combine Equilla functions and perform arithmetic operations. An editor is not necessary.

To apply the following examples, an instrument has to be open in a chart.

Example 1: Calculate a five day average of the trading volume and then the ten day momentum:

```
Momentum( Average( Volume, 5 ), 10 )
```

Example 2: Calculate the absolute difference between the open and close, then the 14 day RSI and then the exponential average for it over 5 days:

```
Drawline( XAverage( RSI( ABS( Open - Close ), 14 ), 5 ) )
```

Example 3: Display the difference between the close and the 200 day average line:

```
Close - Average( Close, 200 )
```

DISPLAY SYNTHETIC TIME LINES, E.G. SPREADS

If you want to display the difference or ratio between two timelines, enter the two symbol shortcuts with the arithmetic operator between them. Instead of entering the symbol short cuts you can also drag the symbol from a symbol list, scanner, watchlist or an open chart to the command line.

For example, enter

```
fdax 2006m / fdax 2006z
```

and choose the option **New Formula Chart** to display the spread.

OPEN WEB ADDRESS

To open a web site, enter a website address in the command line and choose the option **New Browser**.

CHANGE THE TRADING PERIOD

This feature is available for every document with an editable time period. Enter one of the following period shortcuts in the command line and choose the option **Set Intra-Bar Period**:

- **m** or **monthly** - Display a monthly period.
- **w** or **weekly** - Display a weekly period.
- **d** or **daily** - Display a one day period. You can add a number, e.g. "2d" for a two day period.
- **h** - Display a period based on hours. A leading number has to be added, e.g. "3h" for a three hour period.
- **m** - Display a period based on minutes. A leading number has to be added, e.g. "5m" for a five minute period.
- **s** - Display a period based on seconds. A leading number has to be added, e.g. "6s" for a six second period.
- **t** or **tick** - Display a one tick period.

!!BANG COMMANDS

The so-called !!Bang commands in Tradesignal work in a similar manner to the traditional commands for command line interpreters. They always have a leading "!!" (2 exclamation marks) and offer you a quick way to access certain software functions.

- **!!** - Enter only the 2 exclamation marks in the command line to open a list of all available Bang!! commands.
- **!!add** - Adds a symbol to an open document in the workspace. Example: "!!add .dax" adds the dax in a Chart, Scanner, Watchlist or Portfolio.
- **!!close** + option - "!!close d" Closes the active document, "!!close w" closes the active workspace and "!!close all" closes all workspaces and documents.
- **!!eval** + formula name - Adds an Equilla formula in the chart. Example: "!!eval open of .dax" adds the timeline for the dax open to a chart.
- **!!evalseries** + formula series name - Adds an Equilla formula series to a chart.

- **!!log** - Opens the Tradesignal log file which contains important information about the program status and error messages.
- **!!new** + type shortcut (+ content) - Opens a new document. For most documents, the content, e.g. a symbol or web address, has to be given. The following shortcuts are available: b - browser, c - chart, f - Equilla function, i - indicator, m - Scanner, o - Portfolio, p - market profile, s - strategy, t - Position Manager, v - Watchlist, w - workspace.
- **!!replace** + symbol shortcut - Exchanges the symbol in the active chart with the symbol given in the command. Example: "!!replace .dax".
- **!!replaceall** + old symbol shortcut + new symbol shortcut - Exchanges the old symbol in all charts with the new symbol. Example: "!!replaceall ADS GER".
- **!!set** + document property - Changes the property settings of a document. Example: "!!set historylength 100" for a chart.
- **!!setall** + document property - Changes the property settings of all documents in which this property is applicable. Example: "!!setall historylength 100" for all charts.

TOOLBAR

In Tradesignal, the classic menu is substituted with a Ribbon-style Toolbar.

- There are multiple tab-pages of toolbar buttons and menus organized according to general functionality.
- The first tab is the **File** menu, that will show the backstage view of the application when selected. The backstage view is an interactive menu that contains all functions that operate on a workspace as a whole (for example printing).
- The **Home** tab contains the majority of functions that pertain to the contents of a workspace.
- The remaining tabs contain the functions implied by their title. **Insert** for commands concerning inserting content into a workspace. **Layout** for workspace layout commands. **Order routing** for functions concerned with order routing.
- The final tab **Editor** will only appear if a text editor is selected in a workspace. This tab has all functions concerned with editing Equilla or HTML source code and debugging Equilla content in charts.
- Some buttons open menus that offer further functions. You can recognize these by the small black arrow below the buttons.
- The arrangement of the buttons in the toolbar changes depending on the monitor or program window size. Therefore, either more buttons or fewer buttons (but with menu functionality) are visible.
- For many functions, you can open a help topic. The help is available via the key **F1** if the tool tip says that there is a help file for this feature. The help topic is then displayed in the internal browser.
- You can toggle the Toolbar with the keyboard shortcut **Ctrl + F1**, by double-clicking on a Toolbar tab or by clicking the small arrow on the far right of the toolbar tabs. When collapsed the toolbar can be temporarily shown by clicking on one of the toolbar tabs.



Toolbar

TOOLBAR TABS AND GROUPS

The functions displayed on a Toolbar tab are further organized into named groups to improve identification. The following groups are available by toolbar tab:

FILE TAB

This tab shows the backstage view, which combines typical file operations like save and open with interactive inline content organized as sub menus:

- **Recent** - List of recently opened workspaces.
- **Print** - Print preview and printing.
- **Save & Send** - Functions to export the current workspace in various ways and formats.
- **Help** - The application Help menu.
- **Options** - All application configuration options.

HOME TAB

- **Edit** - The typical clipboard functions like copy and paste.
- **Strategies** - Functions to add and augment trading system components.
- **Table** - Functions concerning table-based elements.
- **Data** - Functions associated with searching for - and manipulating - financial data.
- **Chart** - Functions concerned with chart document.
- **Chart Tools** - Drawing tools for use in charts.
- **Favorites** - Any indicators with short codes that have been color-tagged will be listed here.
- **Style** - Operations for creating and applying styles and templates to workspace elements.
- **Find** - Search for items withing open workspace elements.

INSERT TAB

- **Charts** - Chart-like items that can be inserted into a workspace.
- **Tables** - Table-based items that can be inserted into a workspace.
- **Trading** - Trading-system related elements.
- **Research** - Research tools.
- **Developer** - Developer specific tools.

LAYOUT TAB

- **Layout** - Functions to change the layout of the current workspace
- **Monitors** - Options to utilize multiple monitors.

ORDER ROUTING TAB

- **Order Routing** - Functions to control and configure automatic order routing.

EDITOR TAB

- **Edit** - The typical clipboard functions like copy and paste.
- **Text Editor** - Functions found in a typical developer text editor
- **Export** - Options to export an Equilla script in various ways.
- **Find** - Typical search (and replace) operations.
- **Debug** - Commands to control the Equilla Debugger.
- **Breakpoints** - Set and manage breakpoints for use when debugging.

QUICK ACCESS TOOLBAR

The Quick Access Toolbar allows you to create a personalized toolbar comprised of buttons and menus found on the various Toolbar tabs. The Quick Access Toolbar can be displayed above or below the main Toolbar by using the option available in the context menu by right-clicking on the Quick Access Toolbar.

ADDING ITEMS

Items may be added to the toolbar either by

- right-clicking on the desired item in the main Toolbar and selecting the option **Add to Quick Access Toolbar**, or
- using the **+** button in the tree of commands found in the options under *File > Options > Personalize > Ribbon Toolbar*

Some items may not be added to the Quick Access Toolbar. These items will have the **Add to Quick Access Toolbar** option disabled in their context menu.

REMOVING ITEMS

Items may be removed from the Quick Access Toolbar by either

- right-clicking on the desired item on the Quick Access Toolbar and selecting the option **Remove from Quick Access Toolbar**, or
- using the **x** button on the desired item in the **Toolbar Items** list found in the options under *File > Options > Personalize > Ribbon Toolbar*

REORDERING AND GROUPING

Items in the Quick Access Toolbar may be reordered and/or grouped using the **Toolbar Items** list in the options under *File > Options > Personalize > Ribbon Toolbar*. The up and down arrows against an item may be used to reposition that item. The **Begin Group** box may be checked to inset a group separator before the selected Toolbar item.

TOOLBOX

The toolbox on the lower right grants you access to many key features of Tradesignal, namely symbol lists, properties, indicators, strategies, (Equilla) functions, templates, and workspaces.

GENERAL FEATURES

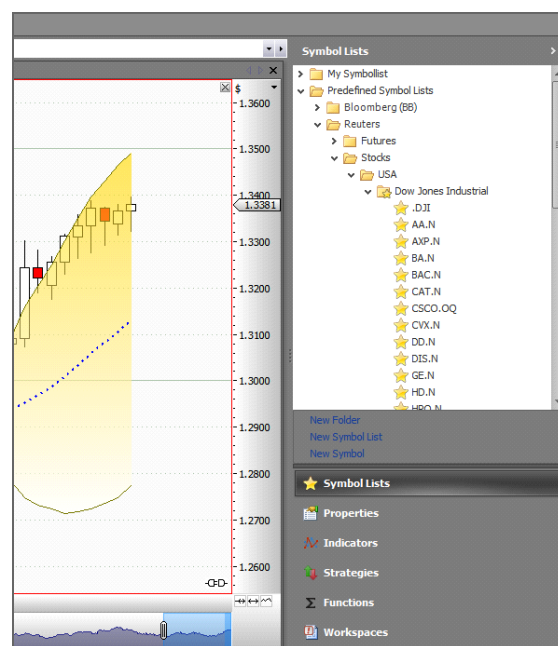
CONTEXT MENUS

In all windows (except for properties) you can right-click on a list entry to open its context menu. There, you can find functions like applying, editing, renaming, deleting, etc. You can also tag list entries with colors to find them more easily later.

RELATED TASKS AREA

Here you can find links to related tasks. These depend on the active tool.

- For symbols, the available links are **New Symbol List**, **New Symbol** and **New Group** (see the chapter Symbol Lists).
- For indicators, strategies, functions and workspaces, you can find the link **New...** (indicator, strategy etc.) and links for Package Management (**New Package** and **Show/Hide Packages**). Packages correspond to folders on your computer and should make tool handling easier for you.
- Templates are saved from charts (see the chapter Save a Template), so only links for Package Management are available here.
- For properties, no links to related tasks are available.



Toolbox

EDIT THE TOOLBOX APPEARANCE

You can also decrease and increase the number of tool buttons, as well as completely removing tools from the toolbox. Click on the small arrow head on the lower right to open the context menu of the toolbox. You have the following options:

- **Show More Buttons** ? Display one of the small buttons below the toolbox as a regular tool button. You can repeat this until all tools are shown as tool buttons.
- **Show Less Buttons** ? Display one of the tool buttons as small symbol button below the toolbox. You can repeat this until all tool buttons are shown as small buttons.
- **Add or Remove Buttons** ? Here you can completely suppress the display of tools in the menu. For this, click on the respective entries in the subordinate menu to switch the tools on or off.

Alternatively you can reduce/expand the display of the tool buttons by dragging the separator between the button menu and the display window upwards or downwards. For this, position the mouse cursor on the dotted grip bar above the tool buttons so that it changes into a double arrow. Then drag the grip bar up or down and release the mouse button when done.

REPOSITIONING TOOLBOX ELEMENTS

Toolbox buttons can be reorganized by dragging and dropping them either

- vertically within the same list of buttons, or
- to the opposite edge of the application window, to split the toolbox on both left and right sides

To drag all toolbox elements to the opposite side, hold the **Ctrl** key while dragging a single button.

COLLAPSING AND TEMPORARILY SHOWING THE TOOLBOX

You can hide and show the toolbox by pressing the small arrow button to the right of the toolboxes title.

To hide and show the properties window directly, press **Ctrl + F2**.

When the Toolbox is collapsed, it can be temporarily shown by clicking on the collapsed toolbox in any location other than the arrow button used to restore the toolbox.

To temporarily show a specific toolbox pane, click the corresponding icon.

DELETING AND RESTORING ITEMS

If a toolbox allows the deletion of its contents, normally using the right-click context menu and selecting **Delete**. The *most recently deleted item* may be restored clicking on the **Unremove** button that is either displayed in a yellow pop-up at the bottom of the application window, or the **Unremove** link shown to the left of the clock at the top right of the application window. Please note, only the most recently deleted item may be restored in this way.

If an item that is deleted has a corresponding file (for example workspace, indicators, strategies, functions and templates), the deleted file will be moved to the system Recycle Bin if it was stored on a file system that supports the Recycle Bin function.

NAVIGATING TO AN ITEM IN WINDOWS EXPLORER

If an item or folder in the Toolbox has a corresponding file (e.g. workspaces and indicators), the right-click context menu for the item or folder will have an option called **Open In Windows Explorer**. Clicking on this option will open the file or folders location in Windows Explorer.

SELECTING MULTIPLE ITEMS AT ONCE

To perform an operation on multiple items at once, first select the items by either:

- holding down the **Ctrl** key and clicking on each item, or
- click on the first item to select, then hold down the **Shift** key and click on the last item to select all items between these two items.

Once multiple items have been selected, the right-click context menu will show only those operations that are applicable to

multiple selected items. Typically allowed operations include: deleting the selected items, moving the selected items between folders, color tagging all the items, or opening the items in various ways within a workspace.

AVAILABLE TOOLS IN THE TOOLBOX

SYMBOL LISTS

Here you can find the list of symbols for the available securities. Read the chapter Symbol Lists for more information.

PROPERTIES

In the properties window, you can access all objects and their properties within the document. The objects are listed at the top of the window. The properties of the selected object are listed below, sorted by category. The properties are discussed in the topic of each object.

INDICATORS

Here you can find all indicators that were delivered in the Tradesignal data folder, as well as all indicators you may have saved in other packages. Read the chapter Indicators for more information. A search box at the top of the pane allows searching by title keywords or indicator short code.

STRATEGIES

Here you can find all strategies that were delivered in the Tradesignal data folder, as well as all strategies you may have saved in other packages. Read the chapter Strategies for more information. A search box at the top of the pane allows searching by title keywords or strategy short code.

FUNCTIONS

Here you can find all functions that were delivered in the Tradesignal data folder, as well as all functions you may have saved in other packages. Read the chapter Functions for more information. A search box at the top of the pane allows searching by title keywords.

TEMPLATES

Here you can find templates, in which graphical properties and information about indicators and trading systems are saved. Via the context menu, you can directly call up a template. It will contain the security data with which it was originally saved. Read the chapter Using Templates for more information. A search box at the top of the pane allows searching by title keywords.

WORKSPACES

Here you can find saved workspaces. Via the context menu you can assign up to ten keyboard shortcuts. Read the chapter

Workspaces for more information. A search box at the top of the pane allows searching by title keywords

HELP

The help is divided into categories that are displayed as main entries. Under most main entries, sub-topics are available. At the top of the help window you can find an input box. Here you can enter a search term for a full-text search in the help. Alternatively, double-click on a list entry to open the topic in the browser window. Via the context menu, you can open the help topic in a new browser window. For this, click on **Open Link in new Browser**. The new browser window is then stacked upon the first one (see also the section Move Documents in the Workspace).

COLOR-TAGGING ENTRIES

Except for symbols and properties, you can color-tag entries in the Toolbox. Color-tagged entries will appear on top of the list, sorted by the three colors yellow, red, and blue.

To assign a color tag, right-click on an entry and select **Yellow Tag** (red, blue) from the context menu.

To remove the tag again, click on **No Tag**.

If a color-tagged indicator or strategy also has a Short Code set, this item will be displayed in the Toolbar in the *Home > Favorites* group. Options to configure this feature can be found in the options under *File > Options > Advanced > Indicators & Strategies*.

KEYBOARD SHORTCUTS

Many functions in Tradesignal can be called up by using keyboard shortcuts. You can find a complete list of all shortcuts in the following tables.

GENERAL APPLICATION

Description	Keyboard Shortcut
Help	F1
Jump to Command Line	F2
Start Symbol Search	F3
Start Print Preview	Ctrl + P
Open / Close the toolbar	Ctrl + F1
Open / Close the property inspector in the toolbox	Ctrl + F2
Delete text, documents, or elements in documents	Del

NAVIGATION IN THE CHART

Description	Keyboard Shortcut
Move up price axis	Arrow up
Move down price axis	Arrow down
Move time axis to the right	Arrow to the right
Move time axis to the left	Arrow to the left
Zoom in price axis	Shift + Arrow up
Zoom out price axis	Shift + Arrow down
Zoom out time axis	Shift + Arrow to the right
Zoom in time axis	Shift + Arrow to the left
Page up price axis	Ctrl + Arrow up
Page down price axis	Ctrl + Arrow down
Page time axis to the right	Ctrl + Arrow to the right
Page time axis to the left	Ctrl + Arrow to the left
Toggle between subcharts	Page up
Toggle between subcharts	Page down
Jump to the beginning of the chart	Pos1
Jump to the end of the chart	End
Magnetic mode for drawing tools	F6
Show standard time span	F8
Show complete time span	Ctrl + F8

WORKSPACES

Description	Keyboard Shortcut
Arrange workspaces	F9
Go to previous workspace	Ctrl + Q
Go to next workspace	Ctrl + W
Toggle between full view (maximized) and normal view	F4
Reload data (like chart, scanner, browser)	F5
Reload data in all workspace elements	Ctrl + F5
Go to next workspace element	Ctrl + Tab
Go to previous workspace element	Ctrl + Shift + Tab
Open new workspace	Ctrl + N
Create new Equilla strategy	Ctrl + Shift + N, S
Create new Equilla indicator	Ctrl + Shift + N, I
Create new Equilla function	Ctrl + Shift + N, F
Open dialog for opening workspaces or packages	Ctrl + O
Save workspace	Ctrl + S
Close active workspace element	Ctrl + F4
Close active workspace	Ctrl + Shift + F4

CLIPBOARD

Description	Keyboard Shortcut
Select all (for text, Equilla code, tables)	Ctrl + A
Copy selected document, text, Equilla code or table element	Ctrl + C
Cut selected document, text, Equilla code or table element	Ctrl + X
Insert clipboard contents into active document	Ctrl + V
Repeat last clipboard action	Ctrl + Y
Undo last clipboard action	Ctrl + Z

PROGRAMMING AND DEBUGGING

Description	Keyboard Shortcut
Compile Equilla code	F7
Toggle comments for the selected code	Ctrl + K, C

Search for text	Ctrl + F
Search and replace text	Ctrl + H
Jump to a specific line number	Ctrl + G
Pause the debugger	Pause
Continue execution	F5
Restart execution	Ctrl + Shift + F5
Step over	F10
Step in	F11
Step out	Shift + F11
Quick watch	Shift + F9
Toggle breakpoint	F9
Enable/disable breakpoint	Ctrl + F9
Open related function	F12
Save indicators, trading systems or Equilla functions	Ctrl + Shift + S
Show all available functions	Ctrl + Space
Show all available variables for a function	Ctrl + Shift + Space

⋮ FINDING AND HANDLING SECURITIES

SYMBOL SEARCH

The symbol search is a useful tool to keep an overview of all the securities available on markets around the world. You can use the sophisticated Tradesignal symbol search for the following:

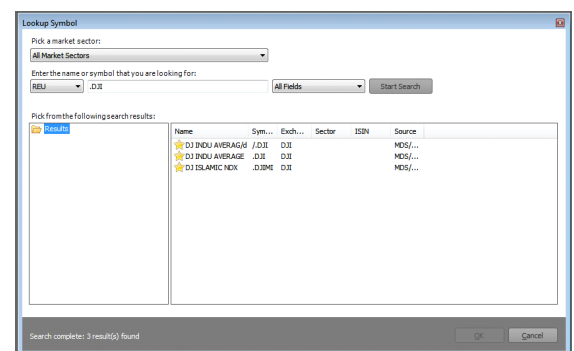
- Searching for unknown symbols
- Assembling symbol lists with securities of similar categories

START THE SYMBOL SEARCH

There are several ways to start the symbol search:

- Click the **Search** button in the toolbar.
- Press the key **F3**.
- Open the context menu of a symbol list and select the entry **New Symbol**, then the option **Standard Symbol**.
- In the toolbox, start a **New Symbol List**, select the option **User-defined Symbol List** and click the **Add** button in the next window.

A search window opens in which you can further define your search terms. Please note that the available search options depend on the used data provider.



Symbol Search

MARKET AND EXCHANGE

By selecting a specific market sector, you can limit the search and result list, for example to avoid finding Wheat Future when you are actually searching for a bond starting with "WHE".

All market sectors like bonds, call options, futures, certificates etc. are available. The standard setting is **All Market Sectors**.

In the **Exchange** list you can find all exchanges offered by your data provider.

SEARCH A NAME OR SYMBOL

Here you can limit the search scope further. Enter the (first) letters of the name or symbol. You can search for the following options (to the right of the entry field):

All Fields - Search all fields (default setting).

Only Name - Search by security name. Example: For "WHE", "Anywhere MD" might be found, although its symbol is "ANWM PNK".

Only Symbol - Search by symbol. Example: For "WHE", "WHE GER" might be found, although the security name is "Pacific Textil Holdings".

Only ISIN - Search by ISIN (International Securities Identification Number).

USE THE SEARCH RESULTS

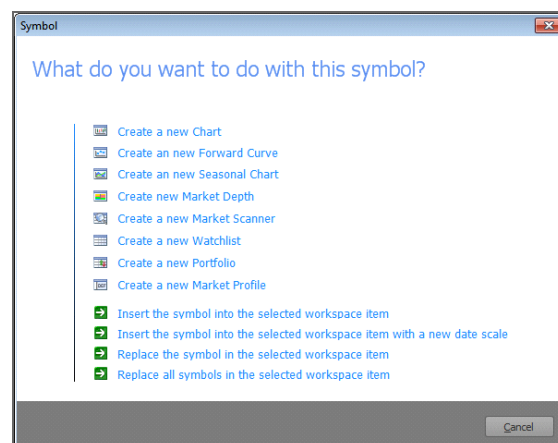
USE THE SEARCH RESULT DIRECTLY

When you start the search with the **Search** button in the toolbar or by pressing **F3**, you can only select a single symbol. You have the choice of actions to perform with the security.

You can open the following new documents: Chart, Market Depth, Scanner, Watchlist, Portfolio, Market Profile.

Alternatively, you can insert the symbol into the selected element of the workspace, e.g. a chart. This can be done in the following four ways:

Insert the symbol into the selected workspace item - Select this to add the found item as a second symbol in a chart, for example.



Symbol Actions

Insert the symbol into the selected workspace item with a new date scale - Select this to add the new symbol, for example as a subchart with its own date scale.

Replace the symbol in the selected workspace item - Select this to substitute the new symbol for the original symbol.

Replace all symbols in the selected workspace item - Select this to substitute the new symbol for the original symbols, for example in several charts.

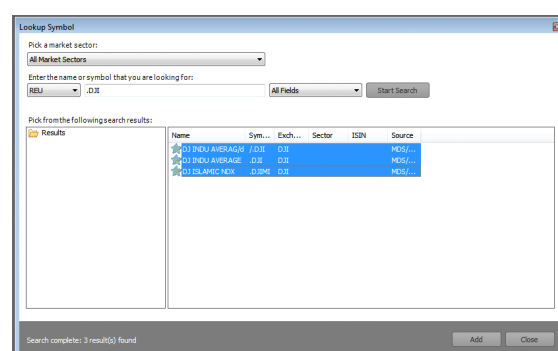
ADD SEARCH RESULTS TO A SYMBOL LIST

If you search via the **Add New Symbol List** link and the option **User-Defined Symbol List**, you can add more than one symbol.

If you fill a new symbol list, you have to enter a name for it.

- Click the **Add** button to add the current result to the symbol list without closing the search window.
- Click the **Close** button to end the search. The new symbols are displayed in the symbol list.

If you set up a new list, confirm the selection with **Done**.



Adding several symbols to a list

EXAMPLE: SET UP A NEW LIST WITH OIL BONDS

1. Start a new symbol list by clicking on **New Symbol List** in the *Related Task* area.
2. Choose the option **User-defined Symbol List**. A new dialog opens.
3. Enter the name *Oil Securities*.
4. Click the **Add** button to start the symbol search.
5. Search securities with the name part "Oil".
6. Select the top symbols from the list by keeping the Shift key pressed while clicking, and confirm with **Close**.

The program returns to the list in the Symbol List wizard. Click **Done** to finish. In the symbol list manager, you can now find the new list "Oil Securities".

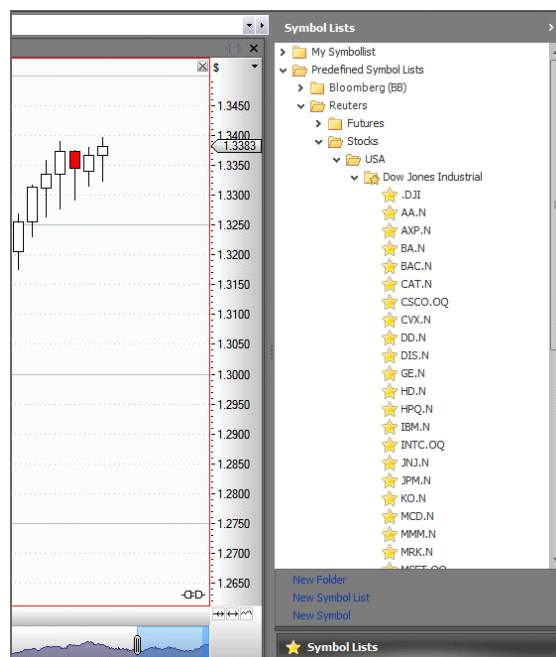
SYMBOL LISTS

Symbol lists include the symbols and names of traded securities. A symbol list corresponds to a category, a branch or index or is assembled by the trader. You can use the symbol lists for the following:

- As an "extended memory" for a high number of symbols
- Sorting symbols according to branches, countries or other criteria
- As a basis for Charts, Scanners and other documents
- For filing symbols from the Scanner or Symbol Search
- For filing formulas or other calculations with which to generate charts

Tradesignal offers a symbol list manager in which you can administrate and sort a large number of lists. You can find it in the toolbox under the button **Symbol List**.

Since single symbols are always managed as part of a symbol list, the special functions for those are also described in this chapter.



Symbol Lists

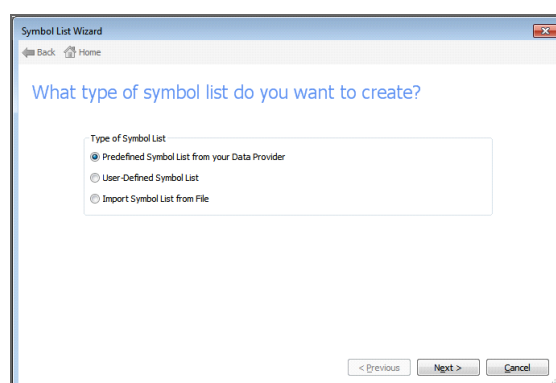
START A NEW SYMBOL LIST

When starting Tradesignal for the very first time, you are asked if you want to receive lists from your data provider. If you confirm this, you will use the Symbol List wizard for the first time. However, you can start the wizard any time to create a new list.

1. In the toolbox, click the **Symbol Lists** button.
2. In the *Related Task* area, click on **New Symbol List**.

The Symbol List wizard opens. Three options are available:

- **Predefined Symbol List from Your Data Provider** - Receive standard lists, for example NASDAQ indexes.
- **User-defined Symbol List** - Start a new, empty symbol list to which you can add symbols or symbol lists, see below.
- **Import Symbol List from File** - Import a symbol list from a file in CSV format (comma separated value) or XML format. The expected separators for CSV in Tradesignal are a comma or semicolon. One way to create files in such a format is to save your data in Excel as CSV text file or use **export** on a symbol list.



Symbol List Wizard

MANAGE SYMBOLS AND SYMBOL LISTS

EDIT VIA THE CONTEXT MENU

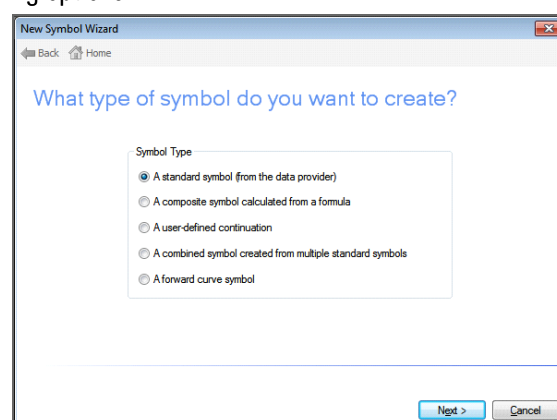
You can manage Symbols and Symbol Lists with the functions offered in the context menus.

- **Open** single symbols in a Chart, Scanner, Portfolio, Watchlist, Market Depth, Market Profile or News.
- **Open** a symbol list in a Scanner, Watchlist or Portfolio.
- You can **open** a symbol from a list in an editor as long as it is a composite symbol.
- You can **add** a new symbol to your existing list.
- A symbol list can be **refreshed**. This option is only available for predefined symbol lists from the data provider. Tradesignal will update changes of the predefined lists after restart of the application, unless you do a manual refresh.
- You can **rename** a symbol or symbol list.
- You can **delete** a symbol or symbol list from the list.
- You can **duplicate** a list. A one-to-one copy will be created, which can be modified and renamed.
- You can **print** a symbol or symbol list. With this function, all information for each symbol is printed. This can take a while if you print a long list of symbols. For more information, see the chapter Publishing.
- Select **Optimize** to start the optimizer for the symbol or symbol list. This starts the Optimizer. In the case of a symbol list, all symbols included in the list will be optimized. The results will be displayed in a table, sorted by total net profit.
- You can also **sort** a symbol list, see the section Sort Symbol Lists.
- You can **export** a symbol list in either CSV or XML format. The XML format supports all symbol types, whereas CSV only supports standard symbols.
- You can **Allow Forward Curves** for a user-defined symbol list to be able to create a Forward Curve from it.

ADD A SYMBOL TO A SYMBOL LIST

Single symbols are always managed in symbol lists. To add a new symbol, open the context menu of a symbol list and select **New Symbol**. The New Symbol wizard opens. You can choose between the following options:

- **A standard symbol (from the data provider)** - This corresponds to the Symbol Search and is the standard way to add a symbol.
- **A composite symbol calculated from a formula** - This way you can add a formula, which refers to other securities, as a symbol itself. You can find more information below in the section Create Composite Symbols.
- **A user-defined continuation** - Here you can set up continuations for futures. You can also add symbols to symbol lists by dragging symbols into them from other lists. For more information see the chapter Continuous Contracts.
- **A combined symbol created from multiple standard symbols**



New Symbol Wizard

This option allows you to create a combination of standard symbols, e.g. the same security traded on multiple exchanges. For more information see section Create Combined Symbols.

DRAG AND DROP SYMBOLS

- You can move single symbols into another symbol list by Drag&Drop.
- You can drag a symbol into a document in the workspace, for example to add the symbol to a Scanner table.
- Drag a symbol into an existing chart to create a subchart or drag a symbol while holding CTRL to replace the chart.
- You can also go the other way round and drag a symbol from an open document (like a Chart, Scanner, Watchlist or Portfolio) into a symbol list. To do this, click into the chart legend or - in case of tables - click into a table cell until a plus symbol and a rectangle appear under the mouse cursor. Then drag the symbol into the symbol list.
- If you drag a symbol to an open editor, its symbol shortcut will be inserted enclosed in single quotes. This can be used for Equilla inline instruments.

DRAG AND DROP SYMBOL LISTS

- Symbol lists can be moved between folders. To do this, simply drag the list into the target folder.
- Symbol lists can also be dragged into an open Scanner, Watchlist or Portfolio. They can also be dragged to an open Chart, this will open all contained symbols.
- If you drag a symbol list to an open editor, its symbol list shortcut will be inserted in double quotes. This can be used for lists of Equilla inline instruments. For further details see the Equilla help in Tradesignal.

SET UP A SYMBOL LIST WITH SEVERAL SYMBOLS

When creating a new symbol list, you can add symbols in the process.

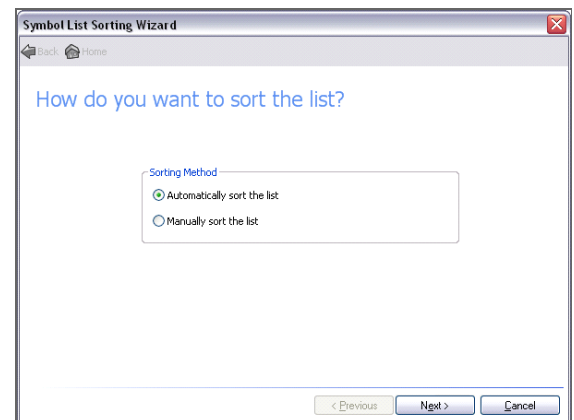
1. In the symbol list manager, click on the folder in which the new list should be created.
2. Select **New Symbol List** in the context menu of the folder. The Symbol List wizard opens. Enter a name for the list and click the **Add** button to start the symbol search for finding and adding symbols.

SORT SYMBOL LISTS

You can sort a symbol list in two ways. Select the entry **Sort Symbol List** in the context menu of a symbol list to start the Symbol List Sorting wizard.

Automatically sort the list - Sort the list automatically using criteria defined by you. The possible sorting criteria are **Ticker Symbol** (default) and **Instrument Display Name** (full name of the symbol, displayed in brackets [] after the symbol). Set a sorting direction of **ascending** (default) or **descending**.

You can also choose the options **Sort numerically** if your securities include numbers, and **Ignore the case of items**. Both options take longer for sorting.



Symbol List Sorting Wizard

Manually sort the list - Select this to sort the list manually. A dialog opens in which you can select symbols and change their position in the list by clicking the **Move up/Move Down** buttons. When finished sorting, click **Done**.

A changed order only affects the current symbol list, not all.

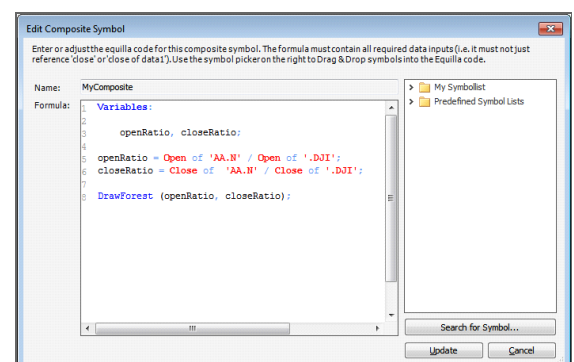
CREATE NEW GROUPS

When you are handling many different symbol lists, it may be useful to structure them further. To do so, create new groups. Click on **New Group** in the *Related Task* area.

CREATE COMPOSITE SYMBOLS

Composite instruments are artificial instruments, calculated from Equilla formulas which refer to other symbols. You can save composite instruments as symbols if the relevant references to symbols and their prices are included in the formulas.

- For a simple composite instrument, a formula can be entered directly into the command line to create a new chart.
- To create a more complex composite instrument, choose **New Symbol** in the context menu of a symbol list and select **A composite symbol calculated from a formula**. Enter the Equilla code and a name. The new composite symbol is added to the current symbol list.



Create a "Composite Instrument" in a symbol list

In Tradesignal you can use the symbol picker to quickly add a symbol to the formula as an inline instrument.

EXAMPLE FOR A SIMPLE COMPOSITE INSTRUMENT

We want to calculate the difference between the open and close of each trading period. The Equilla code for this is

Open - Close

Since we want to save the formula as a composite symbol in a symbol list, references to a security have to be included in the formula, for example .DJI (Dow Jones Industrial). The resulting code is

Open .dji - Close .dji

Enter this formula in the Command Line and select the option **New Formula Chart** in the drop-down menu. The formula is instantly calculated and ready for use. Save the formula as a symbol by dragging its name from the chart legend into a symbol list.

EXAMPLE FOR A MORE COMPLEX COMPOSITE INSTRUMENT

The following code would result in a composite instrument that includes the spread between the Intel symbol and the Dow Jones Industry index.

```
Variables: openRatio, closeRatio;

openRatio = Open of 'intc nas' / Open of '.dji';
closeRatio = Close of 'intc nas' / Close of '.dji';

DrawForest( openRatio, closeRatio );
```

Copy this code into the editor window of a composite instrument, enter a name, and generate the symbol. Afterwards you can display the spread by calling up the composite symbol.

CREATE COMBINED SYMBOLS

Combined symbols are synthetic instruments created by combining the input values of an ordered list of standard symbols and cumulated to a target period from a specified base period.

A combined symbol is very useful when you have a similar product traded at multiple different sources (e.g. Forex and Commodity Brokers) and you would like a combined view of the whole market for that product.

When a combined symbol is created, a base period must be specified, the data from this base period for each of the component symbols will be combined to make the new instrument. Obviously, for periods other than tick there is likely to be some overlapping of data; in this case the values from the component symbol that is specified earlier in the list will be used.

It is recommended to only use tick as a base period when absolutely necessary. The calculation time of the composite will be excessively long and it is unlikely there will be a suitable quantity of tick data to create a daily chart of reasonable length.

Edit a combined symbol

Specify the name of the new symbol, its input symbols and the source period. The order of the symbols is used to select an output value if the timestamps match. To change the field of an instrument, click in the field column and select a value.

Enter the symbol for this instrument (no spaces):
MyCombined

Enter the display name:
MyCombined

Select the instruments that will be combined:

Instrument	Field
.DJI <D> INDU AVERAGE >	default
/.SPX <S&P 500 INDEX/d>	default

Up
Down
Add
Remove
Edit

Select the source period from which the history will be built:
1 Second(s)

OK Cancel

Create a combined symbol dialog

SCANNER

A Scanner (also sometimes called screener) is used for searching large symbol lists according to defined criteria. You can use the Tradesignal Scanner for:

- Searching certain indicator constellations in symbol lists
- Searching for chart patterns in symbol lists
- Searching for technical analysis patterns in price lists
- Backtest of trading systems for several symbols
- Obtaining statistical data like annual high in price lists
- Optimizing trading systems based on a group of symbols

The easiest application, for example, is using the Scanner to search a certain constellation of indicators. Use a relative strength indicator (RSI) > 70 to find the strongest symbols in a list.

Name	Last	Period	Chg	%Chg	Last Trade Date/Time	RSI
DJ INDU AVERAGE	15179.85	Daily	0	0	6/18/2013 3:02:26 PM	46.85
ALCOA INC./d	8.1100	Daily	-0.0100	-0.1232	6/17/2013 10:00:00 PM	29.6314
AMER EXPRESS C/d	73.8400	Daily	+0.8700	+1.1923	6/17/2013 10:02:00 PM	47.2399
BOEING CO./d	103.0300	Daily	+1.2000	+1.1784	6/17/2013 10:00:00 PM	60.1619
BANK OF AMERICA/d	13.2100	Daily	+0.1400	+1.0712	6/17/2013 10:00:00 PM	50.2383
CATERPILLAR IN./d	83.9200	Daily	+0.0500	+0.0596	6/17/2013 10:00:00 PM	33.8452
CISCO SYSTEMS	24.70	Daily	+0.61	+2.53	6/17/2013 10:00:04 PM	67.48
CHEVRON/d	121.2200	Daily	+0.9400	+0.7815	6/17/2013 10:00:00 PM	38.9035
DU PONT CO./d	53.7000	Daily	+1.0200	+1.9362	6/17/2013 10:00:00 PM	40.5375
WALT DISNEY CO./d	64.4900	Daily	+0.6900	+1.0815	6/17/2013 10:01:00 PM	47.5804
GENERAL ELEC C/d	23.7700	Daily	+0.2500	+1.0629	6/17/2013 10:00:00 PM	52.4734
HOME DEPOT INC./d	76.1400	Daily	-0.4500	-0.5875	6/17/2013 10:00:00 PM	39.2254
HEWLETT PACKAR/d	25.1600	Daily	+0.4200	+1.6977	6/17/2013 10:00:00 PM	52.7967
INTL BUS MACHI/d	203.0400	Daily	+0.8400	+0.4154	6/17/2013 10:00:00 PM	44.1507
INTEL CORP	25.10	Daily	+0.18	+0.72	6/17/2013 10:00:04 PM	63.00
JOHNSON JOHNSO/d	85.6300	Daily	+0.7200	+0.8480	6/17/2013 10:00:00 PM	43.7475
JPMORGAN CHASE/d	53.8500	Daily	+0.7200	+1.3552	6/17/2013 10:00:00 PM	52.5934
COCA-COLA CO./d	40.6800	Daily	+0.3400	+0.8428	6/17/2013 10:00:00 PM	42.1496
MCDONALD'S COR/d	98.7400	Daily	+0.3200	+0.3251	6/17/2013 10:00:00 PM	42.9091
3M COMPANY/d	111.9100	Daily	+0.8800	+0.7926	6/17/2013 10:02:00 PM	55.9426
MERCK & CO./d	47.8000	Daily	-0.1500	-0.3128	6/17/2013 10:00:00 PM	52.7108
MICROSOFT CP	34.99	Daily	+0.59	+1.72	6/17/2013 10:00:04 PM	57.58
PFIZER INC/d	29.1500	Daily	+0.0700	+0.2406	6/17/2013 10:00:00 PM	50.7101
PROCTER & GAMB/d	78.9600	Daily	+0.9300	+1.1918	6/17/2013 10:00:00 PM	51.2679
AT&T/d	35.7600	Daily	-0.1500	-0.4177	6/17/2013 10:00:00 PM	41.3170
THE TRAVELERS /d	83.2300	Daily	+1.0800	+1.3147	6/17/2013 10:00:00 PM	49.3769
UNITEDHEALTH G/d	64.3700	Daily	+0.5700	+0.8934	6/17/2013 10:00:00 PM	60.5953
UNITED TECH CP/d	94.9800	Daily	+0.9600	+1.0211	6/17/2013 10:00:00 PM	48.6047
VERIZON COMMS/d	50.7100	Daily	-0.3600	-0.7049	6/17/2013 10:00:00 PM	43.1493
WAL-MART STORE/d	74.9500	Daily	+0.0800	+0.1069	6/17/2013 10:00:00 PM	41.6256
EXODIN MOBIL/d	91.5100	Daily	+0.9300	+1.0267	6/17/2013 10:01:00 PM	49.2628

Scanner

The document type scanner has the following design parameters:

- The maximum history length is 5000.
- The maximum number of instruments is 5001.
- The maximum number of indicators/strategies is 32.
- Indicators and strategies are processed sequentially, i.e. they are applied in the given sequence to each instrument, before the next instrument is processed.

USING THE SCANNER

You can fill the Scanner with symbols in various ways.

WITH THE WIZARD

When choosing one of the following methods, the wizard will open:

- Click on the **Scanner** button in the *Insert* tab of the toolbar.
- In the file menu, select **New** (key shortcut **Ctrl+Shift+N**) to open the Create Item Wizard. Select **Market Scanner**.
- Open the context menu of a symbol list in the toolbox and choose the entry **Open in Market Scanner**. (If selecting this for a single symbol, it opens without a wizard.)

The Create Item wizard opens, in which you can select indicators and strategies in two more steps. Click on **Done** to close the wizard and save the settings.

ADDING ONE OR MORE SYMBOLS MANUALLY

You can also add single symbols or open a new market scanner without using the wizard.

- Open the context menu of a symbol or symbol list in the toolbox and choose the entry **Insert Symbol** to add it to the active

market scanner.

- In the command line, enter a symbol shortcut, e.g. "DD NYS" and select **Add Symbol** to add it to a scanner, or **New Market Scanner** for a new one.
- Via the symbol search. For this, click on the search button and select **Insert the symbol into the selected workspace item** to add it to the scanner, or **Create a new Market Scanner** for a new one.
- You can also add symbols to a scanner via drag&drop from the toolbox or other documents.

The **History Length** (candles/bars) that is regularly taken into account is 2. When you add an indicator, the length may be increased if necessary for the indicator calculation. E.g., the Chande Momentum Oscillator sets the length to 16. If you go below this necessary length, indicators may not give results anymore. In this case, increase the length again.

When you select a symbol list (e.g. the Nasdaq), please ensure that the index itself is not included in the Scanner. It makes no sense to include an index in a list of symbols when scanning for optimal trading parameters. (It is different if you trade in certificates and the scanner is filled with index values.)

SCANNER PROPERTIES

In the properties manager, you can edit the scanner properties. Important parameters are:

History Length - Enter the length of the history here (max. 5000).

Min. Referenced Bars - Enter the minimum number of bars that should be used for calculating the values of indicators and strategies.

SAVING, RESTORING AND RESETTNG DEFAULT SETTINGS

In the appearance area, you can find two important buttons.

- Disk button (Save as Default) - Click here to save your current settings as the new default settings.
- Restore button (Restore Default Settings, circled arrow) - Click here to reset all settings to the default settings.

Not all settings can be saved like this. E.g., the standard period and other user interface settings are entered in the Tradesignal Options.

To restore the original default settings of your Tradesignal installation, click on the **Restore Default Settings** button in the Tradesignal Options.

DELETING SYMBOLS FROM A SCANNER

You can delete symbols from a scanner.

1. Select the symbols. You have the following options:
 - Press **Ctrl+A** to select all.
 - Press **Ctrl** and click on single instruments to select them.
 - Press **Shift** and click on two instruments to select the instruments between them..

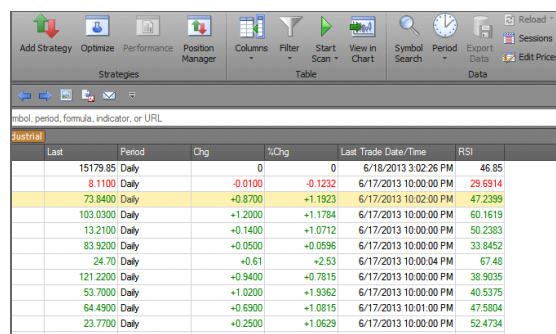
- Press the **Del** key to delete the selected instruments.

SCANNER BUTTONS IN THE TOOLBAR

Start Scan - Start the Scanner.

Stop Scan - Stop the scanning process. All table fields filled at this point are saved.

Period - Here you can set the period that is to be used in the Scanner (this corresponds to the period settings for a chart). The symbols in the Scanner are always called up with the default period at first.



Last	Period	Chg	%Chg	Last Trade Date/Time	RSI
15179.85	Daily	0	0	6/18/2013 3:02:26 PM	46.85
8.1100	Daily	-0.0100	-0.1232	6/17/2013 10:00:00 PM	29.6914
73.8400	Daily	+0.8700	+1.1923	6/17/2013 10:02:00 PM	47.2399
103.0300	Daily	+1.2000	+1.1784	6/17/2013 10:00:00 PM	60.1619
13.2100	Daily	+0.1400	+1.0712	6/17/2013 10:00:00 PM	50.2383
83.9200	Daily	+0.0500	+0.0596	6/17/2013 10:00:00 PM	33.8452
24.70	Daily	+0.61	+2.53	6/17/2013 10:00:04 PM	67.48
121.2200	Daily	+0.9400	+0.7815	6/17/2013 10:00:00 PM	38.9035
53.7000	Daily	+1.0200	+1.9362	6/17/2013 10:00:00 PM	40.5375
64.4500	Daily	+0.6900	+1.0815	6/17/2013 10:01:00 PM	47.5804
23.7700	Daily	+0.2500	+1.0629	6/17/2013 10:00:00 PM	52.4734
76.1400	Daily	-0.4500	-0.5876	6/17/2013 10:00:00 PM	38.7264

Scanner buttons in the toolbar

Columns - Open the menu for the column settings of the Scanner.

Set Group - Here you can group the parameters for better overview. Select a number of rows in the table and click on the button. You can now enter a name for the group. A button appears on top of the table that allows you to open/close the group views like a folder tree in Windows Explorer.

COLUMNS IN THE SCANNER

The table shows different columns depending on the indicators and strategies you are using. The number of columns added for each indicator or strategy depends on the number of output data.

- For each indicator, all columns with a non-static output are displayed. For example, for the "Bollinger Band" indicator there are three columns, for the "Elder Ray" indicator two columns (and two invisible columns with static values).
- For strategies, the two columns "Position" and "Position Size" are displayed over all strategies. The parameters of the single strategies are usually set to invisible in the Equilla code ("visuals" are "inactive"). If they were displayed, they might also interfere with the statistics, which is usually not desired. When at least one strategy is available, you can add columns for statistical output like "Total Net Profit".

SORTING THE COLUMNS

You can sort the columns in the scanner by clicking on the column header. A little triangle appears, pointing up or down depending on the sorting direction. To change the direction, click on the column header again.

Alternatively, right-click into the column and select the sort direction in the context menu.

COLUMN GROUPING

This function in the *Columns* button menu leads to a new row in the table top. If groups are already defined, a button appears for them.

You can start a group by dragging a table row in this new row at the table top, or by selecting a row and clicking on **Set Group** in the toolbar under *Home > Table > Columns*.

ADD/REMOVE COLUMNS

With this function in the *Columns* button menu, you can set columns in the table to visible or invisible.

In the list, all available columns are sorted by categories. Standard columns like price or symbol information are available; so are a large number of statistical values for trading systems. If an indicator offers additional information, it appears as an entry here too.

Select the entries for all columns you want to display, or click on **Show All** to select all entries.

MANAGE STRATEGIES AND INDICATORS

With this function in the *Columns* button menu, you can

- set the processing order of the selected indicators and strategies. To do so, select an entry and click the **Move up/Move down** buttons. The indicators and strategies are applied in this order to every single symbol (sequential processing).
- remove indicators or strategies from the scanner by selecting them and clicking the **Remove** button.

FORMAT OPTIONS

In the Scanner, the following buttons are available in the toolbar.

Styles - Here you can set the display options of the table, e.g. gray background with orange and yellow text.

Sessions - Here you can set the trading times. You can find more information in the chapter Sessions, Holidays and Properties.

Price Editor - Here you can manually edit prices of symbols. You can find more information in the chapter Price Editor.

SCAN RESULTS

The Scanner fills the table with the results from the indicators for all symbols in the table. This may take awhile. If you stop the scanning while it is in progress, the results up to that point are saved and available in the table.

FILTER THE RESULTS

To get a better overview of important results, you can filter the scan result display.

Click on the button **Edit Filter** in the toolbar under *Home > Table > Filter* to open the menu for the filter criteria. There, you can set up e.g. a filter with which you search for an RSI >1 to find the strongest symbols.

You can find more information in the chapter Filter.

USING THE RESULTS WITH DRAG&DROP

USE SYMBOLS IN ANOTHER TABLE

You can reuse symbols from a Scanner in a Watchlist, a Portfolio or another Scanner.

1. If you want to start with a new table, open one by clicking the button **Scanner** (or Portfolio or Watchlist) in the **Insert** tab of the toolbar and closing the wizard with **Done**. An empty table is opened. Alternatively, you can use existing tables.
2. In the Scanner, click into a table cell of the symbol you want to reuse. A little plus sign and a rectangle appear under the mouse cursor. Now drag the cursor into the other table. The symbol is copied into the table.

In a similar way, you can drag symbols from a Portfolio or Watchlist into a Scanner.

SAVING SYMBOLS IN A SCANNER AS SYMBOL LIST

From within the Scanner, you can save interesting symbols as a new symbol list.

1. To save the symbols as a new list, click on the entry **New Symbol List** in the *Related Tasks* area of the symbol manager in the toolbox. Choose the option **User-Defined Symbol List** and name the list but keep it empty.
2. In the Scanner, click into a table cell of the symbol you want to reuse. (By pressing **Shift** or **Ctrl** while clicking with the mouse, you can select more than one row.) A small plus sign and a rectangle appear below the cursor.
3. Since a first click opens the properties tab, click again on the button **Symbol Lists** in the toolbox to open the symbol manager.
4. Drag the selected symbols to the new symbol list. The symbols are added to the list.

OPEN A CHART FROM THE SCANNER AND PAGE THROUGH CHARTS

To view the chart for a symbol from within the Scanner, open the context menu of the table row and select **Open** for the entry **Chart**. The Chart is drawn with the history length and period of the Scanner and includes all indicators and strategies you have set up in the Scanner. The Chart will be linked automatically to the Scanner via symbol linking.

If you want to go through the whole list of symbols in a Scanner to see each chart, use the small, green arrow buttons to the right of the Command Line. As soon as you open one chart of a Scanner entry, you can page through the full list of symbols this way.

EXPORTING SCANNER DATA

The table contents of the Scanner can be copied to other programs. For more information, see the chapter Publishing.

EXAMPLE: HOW TO FIND THE STRONGEST VALUES IN THE NASDAQ

The following parameters can be used for finding the strongest Nasdaq values with the Scanner.

Symbol list:

- All symbols of the **NSD.X** (take care that the index itself is not part of the list)
- **History Length** in the chart properties to "500"

Indicator:

- **Relative Strength Levy** with **Period** 100 days

Start the scanner for the symbols. After the scanning process, sort the table by the RSL column by clicking the RSL column header (if necessary, repeat until the sorting direction is correct). At the top, the strongest Nasdaq value should be displayed.

WATCHLIST

The Watchlist in Tradesignal offers various functions for monitoring stocks in realtime. It is especially helpful for monitoring "ticking" instruments, i.e. instruments with currently changing quotations. Delayed instruments will also be updated automatically.

Watchlists are useful for:

- monitoring symbol lists for their current price development
- monitoring symbol lists for certain constellations of indicators and trading systems
- receiving alerts via mail or acoustic signal when pre-defined triggers are set off

Symbol	Last	Period	Chg	%Chg	Last Trade Date/Time
3M COMPANY/4	111.9100	Daily	+0.8800	+0.7925	6/17/2013 10:02:00 PM
ALCOA INC/4	8.1100	Daily	-0.0100	-0.1232	6/17/2013 10:00:00 PM
AMER EXPRESS C/4	73.8400	Daily	+0.8700	+1.1923	6/17/2013 10:02:00 PM
AT&T/4	35.7600	Daily	-0.1500	-0.4177	6/17/2013 10:00:00 PM
BANK OF AMERICA/4	13.2100	Daily	+0.1400	+1.0712	6/17/2013 10:00:00 PM
BOSW/4	103.0300	Daily	+1.2000	+1.1784	6/17/2013 10:00:00 PM
CATERPILLAR IN/4	83.9200	Daily	+0.0500	+0.0596	6/17/2013 10:00:00 PM
CHEVRON/4	121.2200	Daily	+0.9400	+0.7815	6/17/2013 10:00:00 PM
CISCO SYSTEMS	24.70	Daily	+0.61	+2.53	6/17/2013 10:00:04 PM
COCA-COLA CO/4	40.6800	Daily	+0.3400	+0.8428	6/17/2013 10:00:00 PM
DU PONT CO/4	53.7600	Daily	+1.0200	+1.9382	6/17/2013 10:00:00 PM
EXXON MOBIL/4	91.5100	Daily	+0.9300	+1.0267	6/17/2013 10:01:00 PM
GENERAL ELEC C/4	23.7700	Daily	+0.2500	+1.0629	6/17/2013 10:00:00 PM
HEWLETT-PACKARD/4	25.1600	Daily	+0.4200	+1.6977	6/17/2013 10:00:00 PM
HOME DEPOT INC/4	76.1400	Daily	-0.4500	-0.5875	6/17/2013 10:00:00 PM
INTEL CORP	25.10	Daily	+0.18	+0.72	6/17/2013 10:00:04 PM
INTEL BUS MACH/4	200.0400	Daily	+0.9400	+0.4754	6/17/2013 10:00:00 PM
JOHNSON JOHNSO/4	85.6300	Daily	+0.7200	+0.8480	6/17/2013 10:00:00 PM
JPMORGAN CHASE/4	53.8500	Daily	+0.7200	+1.3552	6/17/2013 10:00:00 PM
MCDONALD'S COR/4	98.7400	Daily	+0.3200	+0.3251	6/17/2013 10:00:00 PM
MERCK & CO/4	47.8000	Daily	-0.1500	-0.3128	6/17/2013 10:00:00 PM
MICROSOFT CP	34.93	Daily	+0.59	+1.72	6/17/2013 10:00:04 PM
PFIZER INC/4	28.1600	Daily	+0.0700	+0.2496	6/17/2013 10:00:00 PM
PROCTER & GAMB/4	78.9600	Daily	+0.9300	+1.1918	6/17/2013 10:00:00 PM
THE TRAVELERS /4	83.2300	Daily	+1.0800	+1.3147	6/17/2013 10:00:00 PM
UNITED TECH CP/4	94.9800	Daily	+0.9600	+1.0211	6/17/2013 10:00:00 PM
UNITEDHEALTH G/4	64.5700	Daily	+0.5700	+0.8934	6/17/2013 10:00:00 PM
VERIZON COMM/4	50.7100	Daily	-0.3600	-0.7049	6/17/2013 10:00:00 PM
WAL-MART STORE/4	74.9500	Daily	+0.0800	+0.1069	6/17/2013 10:00:00 PM
WALT DISNEY CO/4	64.4900	Daily	+0.6900	+1.0815	6/17/2013 10:01:00 PM

Watchlist

With the Tradesignal Watchlist, you get an overview of symbol data in easily manageable tables. Starting from them, you can open symbol charts and page through them for the whole symbol list. The cells with the latest changed values are displayed with colored background.

The document type Watchlist has the following design parameters:

- The maximum history length is 5000.
- The maximum number of instruments is 501.
- The maximum number of indicators/strategies is 32.
- Indicators and strategies are processed sequentially, i.e. they are applied in the given sequence to each instrument, before the next instrument is processed.

WATCHLIST SETUP

You can open a Watchlist with symbols in various ways.

WITH THE WIZARD

When choosing one of the following methods, the wizard will open:

- Click on the **Watchlist** button in the *Insert* tab of the toolbar.
- In the file menu, select **New** (key shortcut **Ctrl+Shift+N**) to open the Create Item Wizard. Select **Watchlist**.
- Open the context menu of a symbol list in the toolbox and choose the entry **Open in Watchlist**. (If selecting this for a single symbol, it opens without a wizard.)

The Create Item wizard opens, in which you can select indicators and strategies in two more steps. Click on **Done** to close the wizard and save the settings.

ADDING ONE OR MORE SYMBOLS MANUALLY

You can also add single symbols or open a new Watchlist without using the wizard.

- Open the context menu of a symbol or symbol list in the toolbox and choose the entry **Insert Symbol** to add it to the active Watchlist.
- In the command line, enter a symbol shortcut, e.g. "DD NYS" and select **Add Symbol** to add it to the active Watchlist, or **New Watchlist** for a new one.
- Via the symbol search. For this, click on the search button and select **Insert the symbol into the selected workspace item** to add it to the Watchlist, or **Create a new Watchlist** for a new one.
- You can also add symbols to a Portfolio via drag&drop from the toolbox or other documents.

The **History Length** (candles/bars) that is regularly taken into account is 2. When you add an indicator, the length may be increased if necessary for the indicator calculation. E.g., the Chande Momentum Oscillator sets the length to 16. If you go below this necessary length, indicators may not give results anymore. In this case, increase the length again.

When you select a symbol list (e.g. the Nasdaq), please ensure that the index itself is not included in the Watchlist. It makes no sense to include an index in a list of symbols when searching for optimal trading parameters. (It is different if you trade in certificates and the Watchlist is filled with index values.)

WATCHLIST PROPERTIES

In the properties manager, you can edit the watchlist properties. Important parameters are:

History Length - Enter the length of the history here (max. 5000).

Min. Referenced Bars - Enter the minimum number of bars that should be used for calculating the values of indicators and strategies.

In the *Indications* area, you can enter the colors for changing values in the table.

Update Indications – Set the background color for table cells with values that changed since the last update, e.g. gray. **Up Move** and **Down Move** – Set the colors for increasing/decreasing values, e.g. green for up, red for down.

SAVING, RESTORING AND RESETTING DEFAULT SETTINGS

In the appearance area, you can find two important buttons.

- Disk button (Save as Default) - Click here to save your current settings as the new default settings.
- Restore button (Restore Default Settings, circled arrow) - Click here to reset all settings to the default settings.

Not all settings can be saved like this.

- The standard period and other user interface settings are entered in the Tradesignal Options.
- In the Watchlist, the order may be subject to changes depending on e.g. changing prices. In the advanced Tradesignal Options, parameter **Keep rows sorted in Portfolio and Watchlist (Secs.)**, you can set the time in seconds after which your original sorting order will be applied to the rows again.

To restore the original default settings of your Tradesignal installation, click on the **Restore Default Settings** button in the

Tradesignal Options.

DELETING SYMBOLS FROM A WATCHLIST

You can delete symbols from a Watchlist.

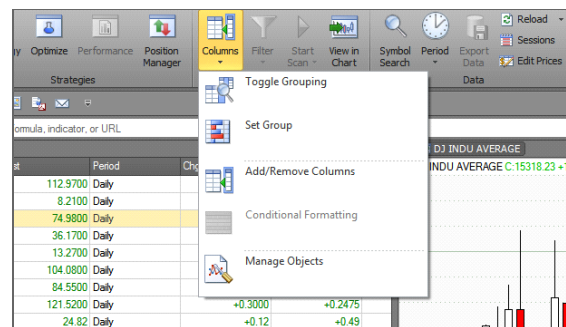
1. Select the symbols. You have the following options:
 - Press **Ctrl+A** to select all.
 - Press **Ctrl** and click on single instruments to select them.
 - Press **Shift** and click on two instruments to select the instruments between them..
2. Press the **Del** key to delete the selected instruments.

WATCHLIST BUTTONS IN THE TOOLBAR

Period - Here you can set the period that is to be used in the Watchlist (this corresponds to the period settings for a chart). The symbols in the Watchlist are always called up with the default period at first.

Columns - Open the menu for the column settings of the Watchlist.

Set Group - Here you can group the parameters for better overview. Select a number of rows in the table and click on the button. You can now enter a name for the group. A button appears on top of the table that allows you to open/close the group views like a folder tree in Windows Explorer.



Watchlist buttons in the toolbar

COLUMNS IN THE WATCHLIST

The table shows different columns depending on the indicators and strategies you are using. The number of columns added for each indicator or strategy depends on the number of output data.

- For each indicator, all columns with a non-static output are displayed. For example, for the "Bollinger Band" indicator there are three columns, for the "Elder Ray" indicator two columns (and two invisible columns with static values).
- For strategies, the two columns "Position" and "Position Size" are displayed over all strategies. The parameters of the single strategies are usually set to invisible in the Equilla code ("visuals" are "inactive"). If they were displayed, they might also interfere with the statistics, which is usually not desired. When at least one strategy is available, you can add columns for statistical output like "Total Net Profit".

SORTING THE COLUMNS

You can sort the columns in the Watchlist by clicking on the column header. A little triangle appears, pointing up or down depending on the sorting direction. To change the direction, click on the column header again.

Alternatively, right-click into the column and select the sort direction in the context menu.

COLUMN GROUPING

This function in the *Columns* button menu leads to a new row in the table top. If groups are already defined, a button appears for them.

You can start a group by dragging a table row to this new row at the table top, or by selecting a row and clicking on **Set Group** in the toolbar under *Home > Table > Columns*.

ADD/REMOVE COLUMNS

With this function in the *Columns* button menu, you can set columns in the table to visible or invisible.

In the list, all available columns are sorted by categories. Standard columns like price or symbol information are available; so are a large number of statistical values for trading systems. If an indicator offers additional information, it appears as an entry here too.

Select the entries for all columns you want to display, or click on **Show All** to select all entries.

MANAGE STRATEGIES AND INDICATORS

With this function in the *Columns* button menu, you can

- set the processing order of the indicators and strategies by selecting an entry and clicking on **Sort Ascending** or **Sort Descending**. The indicators and strategies are applied in this order to every single symbol (sequential processing).
- delete indicators and strategies by selecting an entry and clicking on **Remove**.

FORMAT OPTIONS

In the Watchlist, the following Format buttons are available in the toolbar.

Styles - Here you can set the display options of the table, e.g. gray background with orange and yellow text. You can find more information in the chapter *Styles*.

Sessions - Here you can set the trading times. You can find more information in the chapter *Sessions, Holidays and Properties*.

Price Editor - Here you can manually edit prices of symbols. You can find more information in the chapter *Price Editor*.

ANALYZE THE WATCHLIST RESULTS

The watchlist shows a table of the results of the indicator analysis over all symbols in the list. The data is displayed in real time. Therefore, it is not necessary to explicitly scan the Watchlist to see the current data.

OPEN A CHART FROM WITH THE WATCHLIST AND PAGE THROUGH CHARTS

To view the chart for a symbol from within the Watchlist, open the context menu of the table row and select **Open** for the entry **Chart**. The chart is drawn with the history length and period of the Watchlist and includes all indicators and strategies you have set up in the Watchlist. The Chart will be linked automatically to the Watchlist via symbol linking.

If you want to go through the whole list of symbols in a Watchlist to see each chart, use the small, green arrow buttons to the right of the Command Line. As soon as you open one chart of a Watchlist entry, you can page through the full list of symbols this way.

USING THE RESULTS WITH DRAG&DROP

USE SYMBOLS IN ANOTHER TABLE

You can reuse symbols from a Watchlist in a Portfolio, a Scanner or another Watchlist.

If you want to start with a new table, open one by clicking the button **Watchlist** (or Scanner or Portfolio) in the **Insert** tab of the toolbar and closing the wizard with **Done**. An empty table is opened. Alternatively, you can use existing tables.

In the Watchlist, click into a table cell of the symbol you want to reuse. A little plus sign and a rectangle appear under the mouse cursor. Now drag the cursor into the other table. The symbol is copied into the table.

In a similar way, you can drag symbols from a Scanner or Portfolio into a Watchlist.

SAVING SYMBOLS IN A WATCHLIST AS SYMBOL LIST

From within the Watchlist, you can save interesting symbols as a new symbol list.

1. To save the symbols as a new list, click on the entry **New Symbol List** in the *Related Tasks* area of the symbol manager in the toolbox. Choose the option **User-Defined Symbol List** and name the list but keep it empty.
2. In the Watchlist, click into a table cell of the symbol you want to reuse. (By pressing **Shift** or **Ctrl** while clicking with the mouse, you can select more than one row.) A small plus sign and a rectangle appear below the cursor.
3. Since a first click opens the properties tab, click again on the button **Symbol Lists** in the toolbox to open the symbol manager.
4. Drag the selected symbols to the new symbol list. The symbols are added to the list.

EXPORTING WATCHLIST DATA

The table contents of the Watchlist can be copied to other programs. For more information, see the chapter Publishing.

EXAMPLE: PARAMETERS FOR A WATCHLIST

The following parameters could be used for a Watchlist for index values:

Symbol list:

- all symbols of the .NDX.X (make sure that the index is not part of the list)

Indicators:

- **Net Change** (Change)
- **Net % Change** (Change in %)

Thus configured, the Watchlist offers all important information at a glance. Save the workspace to keep the Watchlist settings.

CONTINUOUS CONTRACTS

User-defined continuations, (UDC) are constructed price time series for futures. Futures are forward contracts offered on basis of commodities, stocks, indexes, currencies or assets. All futures have a fixed expiration date. Therefore, the history length of futures is limited and ends on the day of expiration. For analysis and testing of trading systems this poses a problem since the available data for each contract is too small.

The solution to this problem is continuous contracts, which include the concatenated price data of expired contracts. You can use continuous contracts for

- long-term analysis of future markets
- searching seasonal trends in future markets
- trading system tests over longer histories

Tradesignal offers several ways to set up continuous contracts. Depending on your data provider, you have the choice between several adjustment and adaptation functions. This way, you can add your own creations to the continuous contracts offered by your data provider. The result and all its included calculations is saved as a symbol, which allows further handling and analysis.

SET UP A USER-DEFINED CONTINUATION

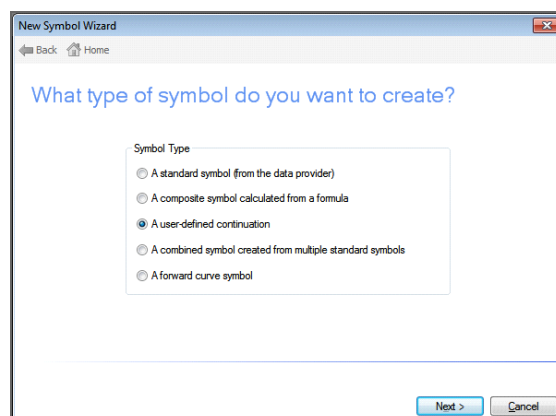
Since a user-defined continuation (UDC) acts like a normal symbol, it is set up via the context menu of a symbol list.

1. Select **New Symbol** in the context menu of a symbol list.
2. Choose the option **User-defined continuation**.

The user-defined continuation wizard opens.

Note that depending on your data provider, some of the following options and adaptation methods might not be available (grayed out).

Setting up an UCD is done in four steps.



User-Defined Continuation Wizard

CONTRACT SELECTION

Future Root Code - Enter a Future Root Code, for example "S" for soybeans.

Starting Contract Year - Enter the starting year for the futures.

Next Contract to Select - If no special specifications have to be met for the contracts, select an entry from the list. However, for several reasons, some contracts are not handled the same way at all times of the year and therefore should be exempted from the history. Therefore there is a way to exempt contracts:

- Exclude the contracts of one or more defined months. For this option, select or deselect the check boxes in front of the months' names. You can also click the **All** button for all months, or the **Quarters** buttons to select contracts from March, June, September and December. Please note that this options only applies to contracts of a monthly expiring nature.
- Exclude the contracts of one or more defined week days. For this option, select or deselect the check boxes in front of the days' names. Please note that this option only applies the contracts that are of a daily expiring nature.

As of version 6.2 of Tradesignal if you use the TMDS as your data provider you can additionally select whether the adjustment calculation should be done on the server or on the client, what until now was the only possible way. Server side calculation is used by default if available because it is faster and less resource consuming. Some options will not be available in this case. Therefore it is possible to switch back to client side calculation by selecting the checkbox **Enable client side calculation options**. All other data providers do not offer this option thus the client side calculation will always be used and cannot get switched off.

ROLLOVER METHOD

Here you can define the rollover method. You can either use fixed intervals or let the market define the rollover moment. For the latter, conditions based on open interest and trading volume are available.

ROLLOVER TRIGGER

Rollover strictly on the expiration day of the contract - The history of the new contract is connected to the history of the expiring contract on the expiration date.

Rollover strictly on a specific day of the expiration month of the contract - For this method, you can define on which day of the expiration month the rollover will take place.

Rollover a number of days before the expiration day of the contract

- For this method, you can define a number of days before the expiration day on which the rollover will take place.

Contract Selection

Rollover Method

Rollover a number of days before the expiration month of the contract - For this method, you can define a number of days before the expiration month on which the rollover will take place.

Rollover when Open Interest of the next contract exceeds the current contract - The rollover date is controlled by the market. When the open interest of the next contract exceeds that of the current contract, the rollover takes place.

Rollover when Volume of the next contract exceeds the current contract - The rollover date is controlled by the market. When the trading volume of the next contract exceeds that of the current contract, the rollover takes place.

Rollover when Open Interest OR Volume of the next contract exceeds the current contract - This is a hybrid of the two previous methods. The rollover takes place when either the open interest or the trading volume of the new contract exceeds that of the current contract. The rollover date is controlled by the market.

Rollover when Open Interest AND Volume of the next contract exceeds the current contract - This is a hybrid of the two previous methods. The rollover takes place when both the open interest and the trading volume of the new contract exceed that of the current contract. The rollover date is controlled by the market.

ROLLOVER TRIGGER OPTIONS

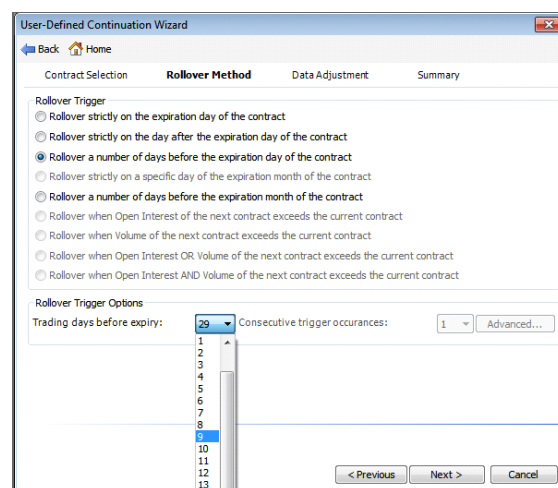
The volume or open interest conditions can be tracked for several days.

Trading days before expiry - Set the number of days for tracking. Select all (*) or the number of days (0-31) from the list.

Consecutive trigger occurrences - Select the number of triggers that have to be set off (1-4).

ROLLOVER - ADVANCED OPTIONS

For some methods, the **Advanced** button is available. It opens another dialog in which you can set rollover intraday details for trading periods shorter than a day.



Rollover Trigger

Use data only from the expiring contract - On the expiration date, only contract data of the expiring contract are taken into account for the calculation.

Use data only from the continuing contract - On the expiration date, only contract data of the new contract are taken into account for the calculation.

Combine data from both contracts into a single candle - The data of both contracts is used to calculate a combined value.

Include non-trading days in all interval calculations - Select this option to include non-trading days into the calculation for intervals of the rollover triggers.

Force rollover to occur at the following time - For trading periods shorter than a day, you can set the time. Enter the hours and

minutes in the respective fields.

DATA ADJUSTMENT

User-defined contracts are constructed by mathematical concatenation of price histories of single contracts. This is done to avoid the so-called rollover gaps, price differences between the expiring and the new contracts. However, this may result in side effects like extreme price divergence in past histories of continuous contracts as compared to the originally traded contracts. Choose the data adjustment method depending on your preferences or to avoid certain side effects.

DATA ADJUSTMENT METHOD

No Adjustment - The data of the contracts is simply concatenated without further adjustments. This results in a large price gap at each rollover date. However, the advantage is that even histories in the distant past still show the real traded prices.

Backward Adjustment - The history of the expiring contract is adjusted to the price of the next contract, i.e. either raised or lowered. For each additionally concatenated contract, the complete past history is adjusted this way. Therefore, the resulting data for the distant past is very different from the actually traded prices. It is also possible to receive negative (nonsensical) values in the process.

Forward Adjustment - The history of the new contract is adjusted to the price of the expiring contract.

Proportional Adjustment - For this, the complete history is calculated anew. All past prices are multiplied with the ratio of the input fields (see below). This way, no negative values can result since the values are multiplied instead of subtracted.

FIELDS WITH WHICH TO CALCULATE THE ADJUSTMENT

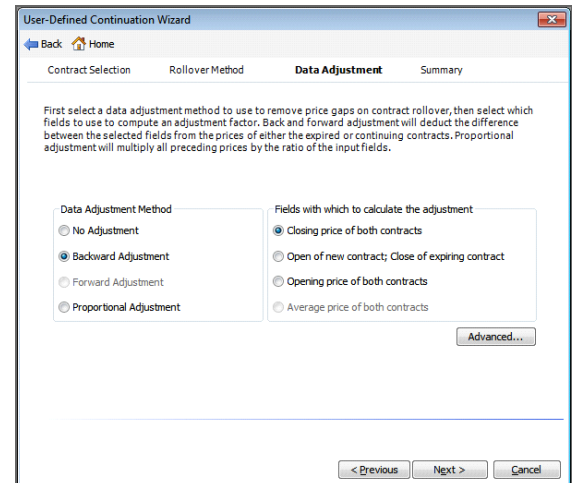
Select which fields the chosen adjustment method should be based on.

Closing price of both contracts - The calculation is based on the closing prices.

Open of new contract; Close of expiring contract - The calculation is based on the close of the expiring and the open of the new contract.

Opening price of both contracts - The calculation is based on the opening prices.

Average price of both contracts - The calculation is based on an average, namely $(\text{Highest High} + \text{Lowest Low}) / 2$ of both contracts.



Data Adjustment

ADVANCED OPTIONS FOR THE ADAPTATION

For some settings, an additional dialog is available under **Advanced....** Here you can define which contract data should be used for calculating the price difference.

The bar after the rollover day

Same day as the rollover day

SUMMARY

In the summary you can see an overview of your settings. You can also select the option **Create a New chart using this UDC when the wizard finishes.**

MANAGE CONTINUOUS CONTRACTS

Via the context menu of the contract symbol you can reach all other functions for the contract:

- You can use continuous contracts in the documents Chart, Market Profile, Watchlist, Scanner and Portfolio.
- You can copy, rename or delete contracts.
- You can open the user-defined continuation wizard to edit the adjustment and calculation methods. To do so, select **Edit Continuation Symbol** in the context menu.

EXAMPLE: CREATING A CONTINUOUS CONTRACT FOR FUTURES

1. In the toolbox, click on **Symbol Lists**.
2. Click on **New Symbol** to open the Symbol wizard.
3. Select **user-defined continuation** and click on **Next**. The user-defined continuation (UDC) wizard opens.

In the UDC wizard:

1. For **Future Root Code**, enter a symbol/shortcut, e.g. "S" for soybeans. (Futures are not automatically available for all account types. Contact Tradesignal for an upgrade, if necessary.)
2. From the list of contract series, select the top entry so that all available contracts will be taken into account.
3. Click on **Quarters** to select the four months in which quarters end.
4. Click on **Next**.
5. As trigger for the concatenation, select **Rollover when volume of the next contract exceeds the current contract**.
6. Click on **Next**.
7. Select **Proportional Adjustment** and **closing prices of both contracts** as fields for the calculation.
8. Click on **Next**.
9. Check the **Create a new chart using this UDC when the wizard finishes** option and click on **Done**.
10. Enter a name for the contract.

The continuous contract now appears as symbol in the symbol list.

LINKS

- www.worldlinkfutures.com
- www.premiumdata.net
- www.futuresknowledge.com
- wikipedia.org

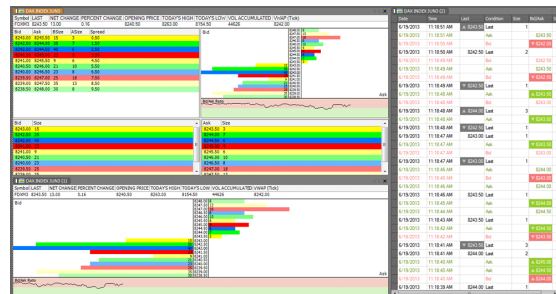
LITERATURE

- Fundamental Analysis - Jack D. Schwager

MARKET DEPTH

The market depth data offer a list of the best bid and ask orders in streaming view and are instrumental for assessing the current state of stocks. This kind of information is also called Level II. It is especially important for day and intraday trading.

- Most of the data and charts in Tradesignal offer **Level I** data, i.e. you receive price information based on the last trade (or ask price, if no trade has taken place).
- In case of **Level II** data, however, you can see current bid and ask orders for a stock. This way, you have an immediate insight into the order books of the traders or ECN. The available details depend on the data provider.



Market Depth

The market depth data give you:

- insight into current bid and ask prices.
- insight into upcoming trades.
- insight into the liquidity of the market.
- the ability to see and judge current trends, accurate to the second.
- the ability to learn about support and resistance price levels (lower and upper limit of price movements).
- insights into the activity of individual market makers (if source given by the data provider).

In Tradesignal, three views are available for market depth analysis:

- **Level II** displays the current status of the order books (available data depends on data provider).
- **Price Ladder** gives a visual overview of the Level II data, as size bars for each offered bid/ask price as well as a bid/ask ratio curve.
- **Time and Sales** displays the current prices, sizes and actual transactions in real time.

OPENING THE MARKET DEPTH VIEWS

1. To open one or all of the available market depth views, click on the button "Market Depth" in the **Insert** tab of the toolbar.
2. Choose a view in the pull down menu or choose "Full Market Depth view" for all.

In principle, you can display market depth data for all traded stocks. However, in the case of rarely traded stocks, the views may remain empty or may be updated infrequently.

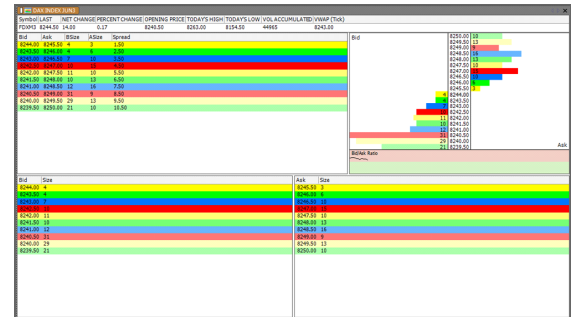
PROPERTIES

- **Display Units:** Here you can set up the way the prices should be displayed (default: 2 decimal places). When you choose fraction display, the new field "Fraction Precision" appears. Here you can set the maximum number of decimal places to be used in case the fraction cannot be expressed by an integral numerator.
- **Layout:** Here you can switch between the available market depth views.

LEVEL II

In the Level II view you can see the bid (buying) and ask (selling) orders and their respective size for the chosen stock. The price levels are color-coded. Besides the price ladder (see below) there are three areas:

- **Overview:** On the upper left, you can find a compressed overview of the bid and ask prices, the respective sizes and the spread between the prices.
- **Bid / Size:** Here you can see the offered bid prices and their size. The best price (highest selling price) is shown on top.
- **Ask / Size:** Here you can see the offered ask prices and their size. The best price (lowest selling price) is shown on top.



Level II

On top of the Level II window, you can find a line with the Level I data (open, high, total, et al.).

The first line of the two areas, with the highest bid and lowest ask, is called "inside market" and traditionally displayed in yellow. The market depth corresponds to the number of visible price levels (default: 10). You can change the default in the properties.

The details of the Level II data depend on the data provider.

PROPERTIES

- **Show Bid/Ask Size:** When this is set to true, the offered sizes are displayed as numbers in the bars of the price ladder.
- **Bid/Ask Ordering:** Allows you to toggle the order of bid and ask in the price ladder (has no effect on the ratio field below the price ladder).
- **Highlight Source:** When this is set to true, orders by the same source in the bid/ask areas are highlighted when clicking on one of the orders (only available if source given).
- **Smoothing:** Here you can enter how many successive ticks should be averaged in the bid/ask ratio to generate a smoother curve in the display.
- **Levels:** Here you can set how many price levels should be displayed in the bid/ask areas.
- **Depth:** Shows the data base used for each line in the overview (e.g. "2*4" for 2 summarized bids and 4 summarized asks).
- **Source:** Displays the individual market maker or ECN in the first column of the bid/ask areas (available data depending on the data provider).
- **Time:** If set to visible, the time column is displayed in the bid/ask areas (available data depending on the data provider).

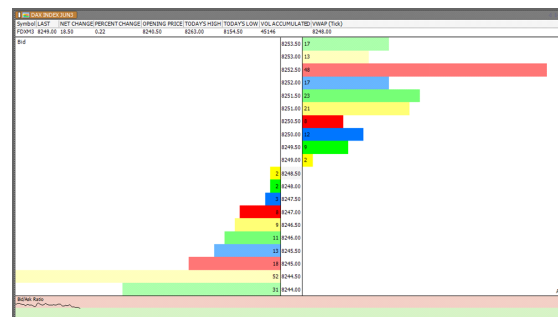
PRICE LADDER

The price ladder offers a visual overview of the Level II data. The price ladder is a regular part of the Level II view but can also be displayed individually. In the case of individual display, a line with the Level I data (open, high, total, et al.) is displayed on top.

The price ladder view consists of:

- The price ladder itself. In it, the sizes of bid and ask orders are displayed as colored bars for each price. This way, you can easily see any existing imbalance between selling and buying orders and also estimate in which direction the price is likely to move. The display order of bid and ask can be set in the properties.
- A curve display for the bid/ask ratio (below the price ladder). The upper red area corresponds to the bid orders, the lower green area to the ask orders. In case of bid overbalance, the curve moves into the red area and prices are likely to move up. In case of an ask overweight, the curve moves down into the green area and prices are likely to drop. A curve on the border between the areas shows a balanced market.

For the price ladder properties, see the section Level II.



Price Ladder

TIME AND SALES

Time and Sales shows a streaming display of price and size of orders and trades as a table. It allows you to see the up and down ticks of the stock price at a glance. Individual market makers or ECN are not available here.

The default color settings of the text are:

- Price changes to best bid price ("To Bid") are shown in red.
- Price changes to best ask price ("To Ask") are shown in green.
- Last, i.e. real trades that correspond to Level I information, are shown in black.
- If the price changes in comparison to the previous bid/ask price, its background color changes accordingly and an additional plus/minus sign is displayed.

Date	Time	Last	Size	Bid/Ask	Condition	Size
6/19/2013	11:22:13 AM	▼ 8249.50	19	8249.50	Bid	
6/19/2013	11:22:13 AM				Last	1
6/19/2013	11:22:12 AM		20	8249.50	Bid	
6/19/2013	11:22:11 AM		19	8249.50	Bid	
6/19/2013	11:22:10 AM		7	▲ 8250.00	Ask	
6/19/2013	11:22:10 AM		18	▲ 8249.00	Bid	
6/19/2013	11:22:10 AM	▲ 8250.00			Last	1
6/19/2013	11:22:10 AM		6	8250.00	Ask	
6/19/2013	11:22:09 AM		7	▲ 8250.00	Ask	
6/19/2013	11:22:09 AM		20	▲ 8249.00	Bid	
6/19/2013	11:22:09 AM	▼ 8249.00			Last	1
6/19/2013	11:22:09 AM	▲ 8249.50			Last	1
6/19/2013	11:22:08 AM		8	▲ 8249.50	Ask	
6/19/2013	11:22:08 AM		3	8248.50	Bid	
6/19/2013	11:22:07 AM		2	▼ 8249.00	Ask	
6/19/2013	11:22:07 AM		4	▼ 8248.50	Bid	
6/19/2013	11:22:07 AM	▼ 8249.00			Last	3
6/19/2013	11:22:07 AM		7	▼ 8249.50	Ask	
6/19/2013	11:22:07 AM		5	8249.00	Bid	
6/19/2013	11:22:07 AM	8249.50			Last	1
6/19/2013	11:22:06 AM		8	8250.00	Ask	
6/19/2013	11:22:06 AM		6	8249.00	Bid	
6/19/2013	11:22:06 AM	▼ 8249.50			Last	1
6/19/2013	11:22:06 AM		9	8249.00	Bid	
6/19/2013	11:22:05 AM		10	8249.00	Bid	
6/19/2013	11:22:05 AM		9	▼ 8249.00	Bid	
6/19/2013	11:22:04 AM		6	▼ 8250.00	Ask	
6/19/2013	11:22:04 AM		1	8249.50	Bid	

Time and Sales

PROPERTIES

- **History Length:** Here you can enter the maximum number of transactions to be stored in the table. Note: Upon changing this value, there might be a noticeable wait while the system updates the historical information.
- **Date:** When this is set to visible, the current date is shown in an additional column.
- **Condition Codes:** When this is set to visible, the conditions are written out in an additional column (e.g., "At Ask").

FILTER

The filter is an important instrument for filtering the results of the optimizer and scanner for certain values.

FILTER VALUES

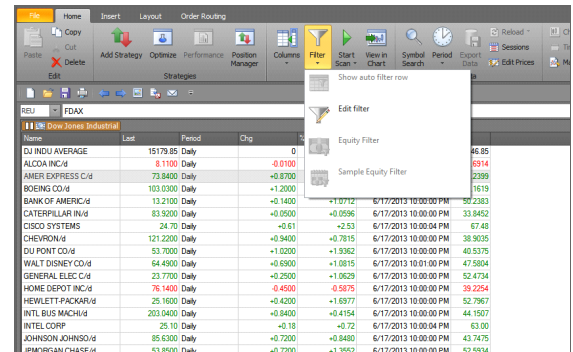
The filtering proceeds in two steps:

1. Click the **Edit Filter** button in the Toolbar under *Home > Table > filter* to set up a filter.

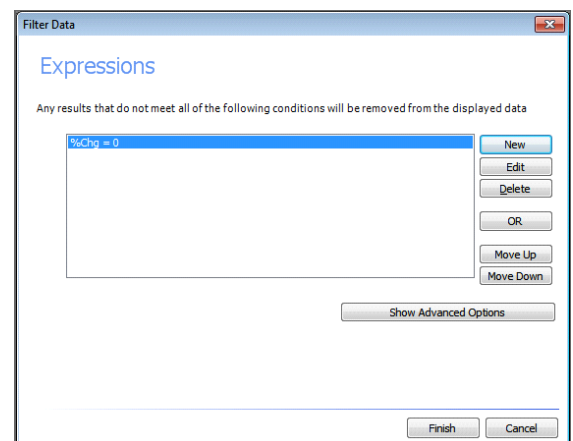
FILTER SET UP

When setting up the filter, you define one or more criteria that form the filter. You have the following options:

- **New** - Starts a new filter. An existing filter will be deleted. It is not possible to set up and save several filters. However, you can enter parallel filters combined by **OR**. A dialog opens. Here you can enter which column should be tested with which operator (=, >, <...) Confirm with **OK**.
- **Edit** - Opens the dialog as explained above, in which you can edit the selected filter criteria.
- **Delete** - Deletes the selected line in the filter.
- **OR** - Choose this to enter other possible criteria sets. This way, you may avoid setting up new filters. Enter all important filters with OR between them. The ones that don't apply are ignored.
- **Move up/Move down** - With these buttons you can move a criterion up and down in the appliance sequence.



Filter buttons in the toolbar



Create a new filter

ADVANCED FILTER OPTIONS

Click on **Show Advanced Options** to display additional options:

- **Show all filtered rows** - All rows of the table are shown. An additional column named **Filter** appears, in which rows are labeled "true" or "not true".
- Alternatively you can combine the row display with another condition by selecting an Equilla formula in the list or entering a new one. In this case, only filtered rows that fulfill the second condition are displayed.

This option is only available for Scanner documents.

EXAMPLE - FILTERING FOR STRONGEST VALUES

1. Open a new scanner and fill it with the 50 symbols of the "Dow Jones Industry", for example. Add the indicator **Relative Strength Levy**.
2. Perform the scan.
3. Click the **Edit Filter** button in the toolbar to open the filter criteria dialog.
4. Click the **New** button and choose the column "RSL" for testing. That's the relative Strength Levy indicator we have entered in the scanner.
5. Select the operator > and enter the number "1" as the value to be tested against. Confirm your settings.

As a result you get a filtered list that includes only stocks with a Relative Strength Levy of more than 1.

CHARTING

CHART TYPES

BAR CHART

Bar charts are among the most common chart types and give information about open, high, low and close prices, price developments and the magnitude of price movements.

The bar chart can be set as the default chart type in Tradesignal, see chapter Chart.

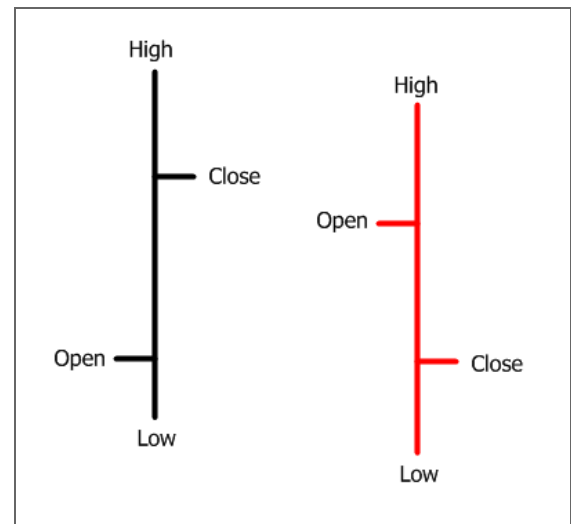
TRADING INFORMATION IN BAR CHARTS

The following information is available in a bar chart:

- The highest price, given by the top point of the bar
- The lowest price, given by the lowest point of the bar
- The open price, given by the left vertical line
- The close price, given by the right vertical line
- The direction of the price movement, given by the relative positions of the left and right vertical lines. If the left one is below the right one, prices have risen during the trading period. Otherwise, prices have fallen.
- The direction of the price movement, given by the color of the bars. The usual color setting is red for falling and black for rising prices. The colors can be edited in the chart properties.
- The strength of the price movement, given by the total length of the bar (difference between high and low).



A Bar Chart



A Single Bar

GENERAL CHART PROPERTIES

When the chart is active, you can edit the chart properties in the toolbox, for example the history length, the axes and legends.

You can also

- set the trading period, see the chapter Timespans and Periods. Note that it is not possible to use bar charts when using ticks as periods.
- change the time span. In this case, several bars of a time span are combined to a "meta bar" to display the trading activity of several periods. For example, if you choose "Current Quarter", three meta bars based on the monthly dates are displayed.
- change the chart look, see the chapter Chart Properties.

CONTENT-DEPENDENT CHART SETTINGS

In the properties of the displayed stock (symbol), you can find additional parameters. Besides the Standard Properties, you can find specific parameters for this chart type.

Bars Properties

Bullish Color - Here you can select a color and transparency for upward bars.

Bearish Color - Here you can select a color and transparency for downward bars.

CANDLESTICK CHART

Starting in the 17th century, the candlestick chart pattern was used in Japan. Used by Charles Dow in 1900, it was the American trader Steve Nison who brought candlestick charts back to the awareness of western traders. Today, candlestick charts are one of the most common chart types.

Candlestick charts give information about the open, high, low and close prices and about the direction and magnitude of the price movement. It is this wealth of information and the distinct patterns that make the candlestick charts so interesting. The direction and movement of prices can be better perceived than by using bars. More complex candlestick patterns are often used as the basis for trading decisions.



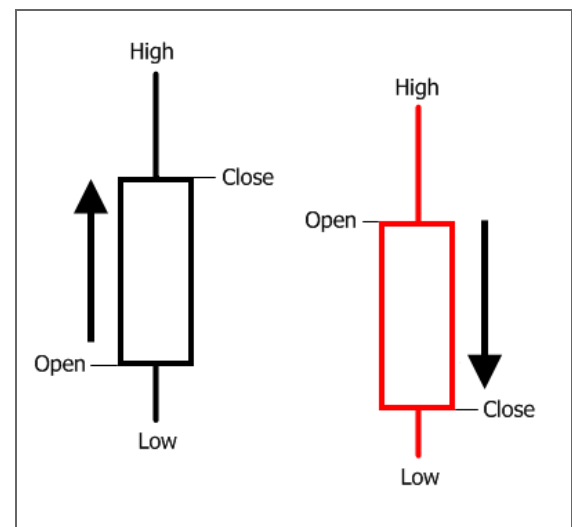
A Candlestick Chart

The candlestick chart is the default chart type in Tradesignal. See also the chapter Chart.

TRADING INFORMATION IN CANDLESTICKS

The following information is available in the candle body and the shadows (the lines on bottom and top, also called wicks):

- The high of the trading period, given by the end of the upper shadow.
- The low of the trading period, given by the end of the lower shadow.
- The open price, given by the body edge; either the bottom line in the case of an upwards development, or the top line in the case of a downwards development.
- The close price, given by the body edge; either the top line in the case of an upward development, or the bottom line in the case of a downward development.
- The direction of the price movement, given by the body color. Colors are usually red or black for falling and green or white for rising prices. The color settings can be changed.
- The magnitude of the price movement, given by the length of the body and the shadows. The total length of a candlestick between the wicks is the difference between high and low.



A Single Candlestick

More information can be gathered when several candlesticks are seen in context. Patterns of up to three candles are used to take a closer look at the price movements and their quality. For example, candles with bodies of the same color show more pronounced trends. Trading is often done based on a mixture of candlestick pattern analysis and other methods.

TWO SIMPLE CANDLE EXAMPLES

THE HAMMER

The hammer was named for its characteristic shape and the relationship to the Japanese sign for "hammering a floor". It has a large lower shadow with a small body on top, meaning a long downward movement stopped (and changed direction) during the trading period.

An ideal hammer has a small white body, showing that a bullish trend won out. The hammer should have no upper shadow, meaning that the close corresponded to the high in the trading period. In combination with analysis methods like moving average and other support signals, a hammer may be a good entry signal.



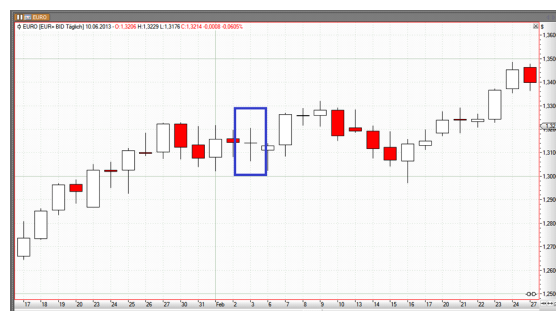
Hammer

THE DOJI

A doji is a common candle that has a very small or no body at all. If the body is small, the color can be ignored. A distinct doji has very large shadows of similar size.

A doji signals that for all the trading over the day, the market and its forces are in equilibrium. The opening and close price are almost the same.

The doji itself only shows a kind of "frozen" state, of indecision. However, it gathers importance depending on the surrounding patterns. Price movements taking place after a doji often signal new trends or a trend inversion. Therefore, dojis are good support signals for resistance and support lines.



Doji

GENERAL CHART PROPERTIES

When the chart is active, you can edit the chart properties in the toolbox, for example the history length, the axes and legends.

You can also

- set the trading period and timespans, see the chapter Timespans and Periods.
- change the chart look, see the chapter Chart Properties.

CONTENT-DEPENDENT CHART SETTINGS

In the properties of the displayed stock (symbol), you can find additional parameters. Besides the Standard Properties, you can find specific parameters for this chart type.

Candles Properties

Border Color - Here you can select a color for the border.

Show Frame - Here you can select whether the candle frame will be displayed or hidden.

Bullish Color - Here you can select a color and opacity for upward candles.

Bearish Color - Here you can select a color and opacity for downward candles.

INDICATORS AND STRATEGIES

In the indicator overview, you can find the indicator **Candle Pattern** which includes a long list of typical candlestick patterns to choose from. You can edit these patterns in the graph properties in the toolbox. The set pattern is displayed in the graph.

In the strategies overview, you can find strategies such as **Candle Bearish Engulfing - Exit**, **Candle Big White Candlestick - Exit**, and many more that you can apply to your candlestick chart.

LINKS AND BOOKS

Candlestick Charts in the Wikipedia

LINE CHART

Line charts are a classic type of chart, in which data points are connected by a line, e.g. physical data in axes of coordinates. The line chart can be set as the default chart type in Tradesignal, see chapter Chart.

In Tradesignal, line charts are used for displaying:

- time lines for symbols, e.g. for Open, High, Low or Close
- indicator results
- performance development of a trading system
- additional information like fundamental data and market depth indices.



Line Chart

The line chart is the simplest chart type for stocks. From trading period to trading period, the line is extended for new prices.

For symbols, the lines are usually drawn for closing prices.

GENERAL CHART PROPERTIES

When the chart is active, you can edit the chart properties in the toolbox, for example the history length, the axes and legends.

You can also

- set the period and time span, see the chapter Timespans and Periods.
- change the chart look, see the chapter Chart Properties.

CONTENT-DEPENDENT CHART SETTINGS

In the properties of the displayed stock (symbol), you can find additional parameters. Besides the Standard Properties, you can find specific parameters for this chart type.

Instrument Properties

Input - Select which data to use for calculating the chart (Open, High, Close, Low, Volume).

Line Properties

Color - Here you can select a color for the line.

Width - Here you can set a line width by entering a number.

Style - Here you can select a line style, e.g. solid or dotted.

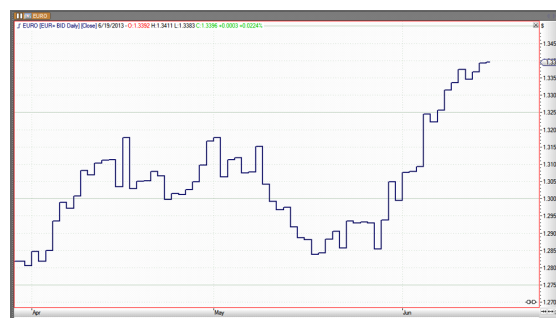
Shadow - Here you can set a shadow for emphasizing the line with a 3D effect. Four settings are available.

STEPPED LINE CHART

The stepped line chart is a variation of the normal line chart. The price levels are connected not by direct lines but at right angles. Stepped line charts can only be used for stocks.

GENERAL CHART PROPERTIES

When the chart is active, you can edit the chart properties in the toolbox, for example the history length, the axes and legends.



Stepped Line Chart

You can also

- set the period and time span, see the chapter Timespans and Periods.
- change the chart look, see the chapter Chart Properties.

CONTENT-DEPENDENT CHART SETTINGS

In the properties of the displayed stock (symbol), you can find additional parameters. Besides the Standard Properties, you can find specific parameters for this chart type.

Instrument Properties

Input - Select which data to use for calculating the chart (Open, High, Close, Low, Volume).

Stepped Line Properties

Color - Here you can select a color for the line.

Width - Here you can set a line width by entering a number.

Style - Here you can select a line style, e.g. solid or dotted.

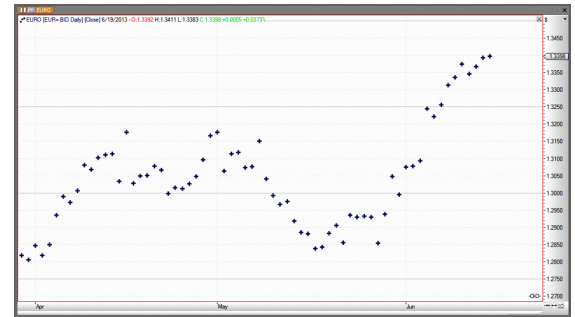
Shadow - Here you can set a shadow for emphasizing the line with a 3D effect. Four settings are available.

SYMBOL CHART

The symbol chart is a variation of a normal line chart, displaying the chosen input (open, close, high...) as unconnected symbols. Several symbol types can be set for the display.

GENERAL CHART PROPERTIES

When the chart is active, you can edit the chart properties in the toolbox, for example the history length, the axes and legends.



Symbol Chart

You can also

- set the period and time span, see the chapter Timespans and Periods.
- change the chart look, see the chapter Chart Properties.

CONTENT-DEPENDENT CHART SETTINGS

In the properties of the displayed stock (symbol), you can find additional parameters. Beside the Standard Properties, you can find specific parameters for this chart type.

Instrument Properties

Input - Select which data to use for calculating the chart (Open, High, Close, Low, Volume).

Dotted Line Properties

Style - Here you can select a symbol style, e.g. circles, squares etc.

Size - Here you can set the symbol size.

Border Color - Here you can select a border color for the symbol.

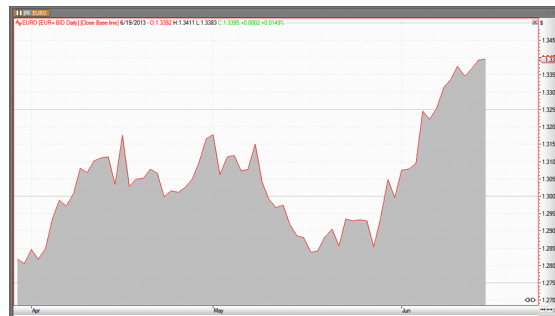
Fill Color - Here you can select a fill color for the symbol.

FILLED AREA CHART

Filled area charts, as indicated by the name, are charts in which filled areas between selected input data is displayed. You can use filled area charts for:

- Drawing a filled area between the prices and the base line (default)
- Drawing a filled area between different price levels

Filled area charts are not useful for detailed symbol analysis, as too much information is lost in this kind of charting.



Filled Area Chart

GENERAL CHART PROPERTIES

When the chart is active, you can edit the chart properties in the toolbox, for example the history length, the axes and legends.

You can also

- set the period and time span, see the chapter Timespans and Periods.
- change the chart look, see the chapter Chart Properties.

CONTENT-DEPENDENT CHART SETTINGS

In the properties of the displayed stock (symbol), you can find additional parameters. Besides the Standard Properties, you can find specific parameters for this chart type.

Instrument Properties

Baseline - Here you can set a value for the price baseline, e.g. raise the baseline to "40".

Input A - Select the data to be used as upper line of the area (Open, High, Close, Low or Volume).

Input B - Select the data to be used as lower line of the area (Open, High, Close, Low, Volume or Baseline).

Filled Area Properties

Color 1 - Here you can select a color for the top line.

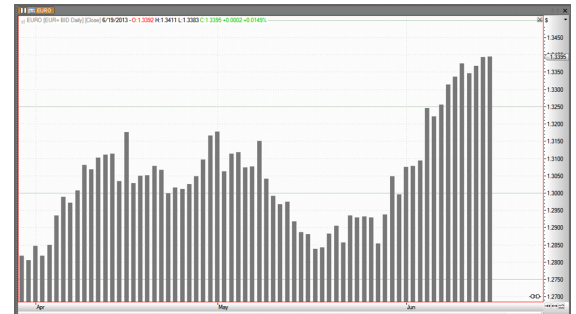
Color 1 - Here you can select a color for the bottom line.

Area Color - Here you can set a color and transparency for the filled area.

FOREST CHART

Forest charts are histograms. For each input value in the timeline, a bar (also called "needle") is drawn.

Forest charts are not useful for detailed symbol analysis, as too much information is lost in this kind of charting. They are more commonly used for indicators, where relative changes per day can be emphasized by needles in different colors.



In Tradesignal, forest charts can be used for:

- symbol prices (by selecting the entry in the chart menu)
- the trading **Volume** (in the stock properties in the toolbox; this is available as an addition for every chart type)
- indicators; however, for those, the equilla scripts have to be adapted. You can find a tutorial for doing so in the chapter [Displaying a Momentum Indicator as Forest Chart](#).

GENERAL CHART PROPERTIES

When the chart is active, you can edit the chart properties in the toolbox, for example the history length, the axes and legends.

You can also

- set the period and time span, see the chapter [Timespans and Periods](#).
- change the chart look, see the chapter [Chart Properties](#).

CONTENT-DEPENDENT CHART SETTINGS

In the properties of the displayed stock (symbol), you can find additional parameters. Besides the Standard Properties, you can find specific parameters for this chart type.

Instrument Properties

Input - Select which data to use for calculating the chart (Open, High, Close, Low, Volume).

Histogram Properties

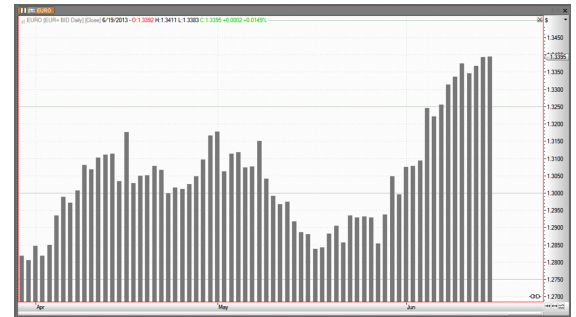
Histogram Color - Here you can select a color for the histogram bars.

TUTORIAL INDICATOR AS FOREST

To display indicators as a forest chart, the indicator code has to be adapted accordingly. In the following tutorial, we show you how to set up a Momentum indicator as a forest chart with the needles in two different colors, depending on their direction.

THE "NORMAL" MOMENTUM INDICATOR

You can find the normal Momentum Indicator in the list of all indicators in the Toolbox. Right-click the indicator to open the context menu and choose **Edit**. The editor with the indicator's equilla code opens.



Forest Chart

Meta:

```
Synopsis( "The Momentum indicator utilizes price and volume statistics for predicting the strength or weakness of a current market and any overbought or oversold conditions, and may indicate turning points within the market." ),
ShortCode( "MOM" ),
SubChart( True );
```

Inputs:

```
Price( Close ),
Period( 10, 1 ),
ShowAlerts( False );
```

```
DrawLine( Momentum( Price, Period ), "MOM" );
DrawLine( 0, "Zero Line", StyleDot );
```

```
CheckForCrossAlert( Plot1, 0, "Momentum", "Zero Line", ShowAlerts );
```

```
// *** Copyright SystemSoft GmbH ***
// *** www.tradesignal.com ***
```

COPY THE MOMENTUM INDICATOR CODE AS BASIS FOR THE NEW INDICATOR

1. Click on **New Indicator** in the *Related Tasks* area of the toolbox to start a new indicator. Call it *Momentum Forest*.
2. Copy the equilla code of the Momentum indicator into the new, empty editor window.

To do so, select the complete code with **CTRL+A**. Then copy the contents with **CTRL+C**. Click into the new, empty editor window and press **CTRL+V** to insert the copied contents. Alternatively, you can find all these functions in the context menu of the editor, which you can open by right-clicking the editor window.

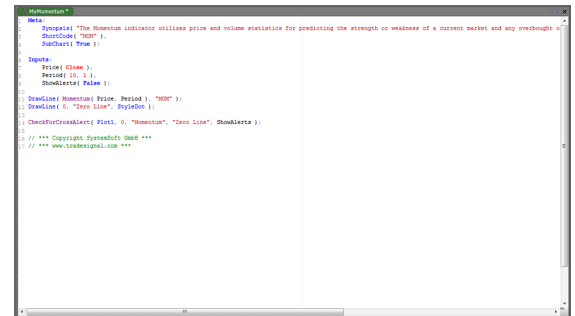
EDITING THE MOMENTUM FOREST INDICATOR FOR DISPLAY AS A FOREST CHART

First, you have to delete some lines in the code we do not want to use:

- `ShowAlerts(False);` (make sure that the line above ends with a semicolon ";")
- `CheckForCrossAlert(Plot1, 0, "Momentum", "Zero Line", ShowAlerts);`

Then you need two variables for your indicator:

- one variable to save the momentum calculation
- one variable to save the color setting for the forest display



HTML Editor with Indicator Code

For this, enter the following lines beneath the *Inputs* area:

```
Variables:
momValue, forestColor;
```

Below this, enter the formula for the Momentum calculation:

```
//Calculate Momentum
momValue = Momentum( Price, Period );
```

Now you need the condition for the needle colors. If the Momentum rises in comparison to the previous day, the needle should be displayed in green; otherwise, the needle should be red.

```
//Set color depending on the change in comparison to the previous day
if momValue > momValue[1] then
    forestColor = DarkGreen
else
    forestColor = Red;
```

To display the result as forest chart, you have to substitute the first `DrawLine` line with the following:

```
//Drawing command with color variable
DrawForest(0, momValue, "Zero", "MomentumUp", thick, forestColor, true);
```

The color of the needle is given by the dynamically set variable *forestColor*. The base line remains unchanged.

In the META area of the code you can now edit the synopsis and enter a specific ShortCode. With the latter, you can later call up your new indicator from the command line.

COMPILING AND APPLYING THE MOMENTUM FOREST INDICATOR

Meta:

```
Synopsis( "The Momentum Forest is a variation of the Momentum Indicator, displaying the
values in Forest fashion with color coding for up/down development. The Momentum indicator
utilizes price and volume statistics for predicting the strength or weakness of a current
market and any overbought or oversold conditions, and may indicate turning points within the
market." ),
ShortCode( "MOMF" ),
SubChart( True );
```

Inputs:

```
Price( Close ),
Period( 10, 1 );
```

Variables:

```
momValue, forestColor;
```

```
//Calculate Momentum
momValue = Momentum( Price, Period );

//Set color depending on the change in comparison to the previous day
if momValue > momValue[1] then
forestColor = DarkGreen
else
forestColor = Red;

//Drawing command with color variable
DrawForest(0, MomValue, "Zero", "MomentumUp", thick, forestColor, true);
DrawLine( 0, "ZeroLine", StyleDot)

// *** Copyright tradesignal GmbH ***
// *** www.tradesignal.com ***
```

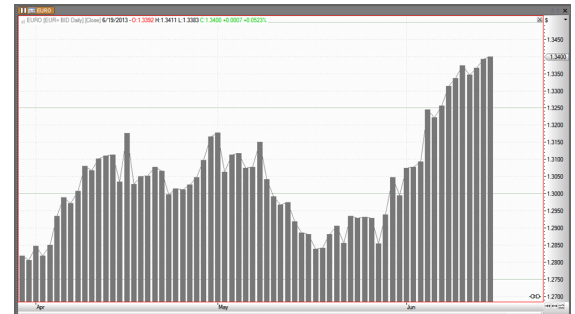
Compile the indicator by pressing the **F7** key or by clicking the **Compile Script** button in the Equilla menu in the toolbar.

The indicator is now available in the indicator list in the toolbox. From there, you can apply it to a chart by either pulling it into a chart via drag&drop or by selecting **Apply** from the context menu.

LINKED FOREST CHART

The linked forest differs only marginally from the Forest Chart. The top of the needles are connected by a line. Other than that, the chart is identical to the forest chart.

This chart is less helpful for symbol charting, as the connection with the baseline can lead to distorted values for longer histories.



Linked Forest Chart

GENERAL CHART PROPERTIES

When the chart is active, you can edit the chart properties in the toolbox, for example the history length, the axes and legends.

You can also

- set the period and time span, see the chapter Timespans and Periods.
- change the chart look, see the chapter Chart Properties.

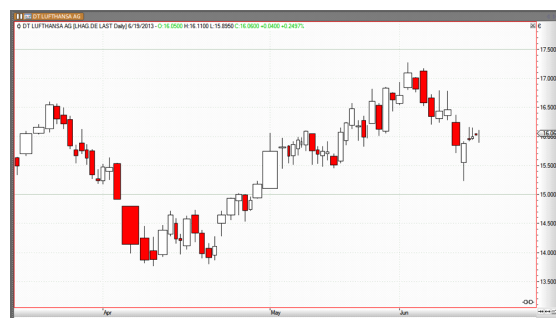
CONTENT-DEPENDENT CHART SETTINGS

You can find the symbol parameters in the properties of the displayed stock (symbol), see the chapter Forest Chart.

CANDLE VOLUME CHART

Candle volume charts combine charts of prices and volumes. The following information is available:

- The high of the trading period, given by the end of the upper shadow.
- The low of the trading period, given by the end of the lower shadow.
- The open price, given by the body edge; either the bottom line in the case of an upwards development, or the top line in the case of a downwards development.
- The close price, given by the body edge; either the top line in the case of an upward development, or the bottom line in the case of a downward development.
- The direction of the price movement, given by the body color. Colors are usually red or black for falling and green or white for rising prices. The color settings can be changed.
- The magnitude of the price movement, given by the length of the body and the shadows. The total length of a candlestick between the wicks is the difference between high and low.
- The trading volume, given by the width of the candle body.



Candle Volume Chart

Candle volume charts are a mixture of candlestick and equi volume charts. The candle interpretation is the same as for Candlestick Charts, but the width of the candle depends on volume, not on time. The difference to Equi Volume Charts is that, as opposed to that chart type, the open and close prices are still available in the candle volume charts.

LIMITATIONS

At each end of the quarter, there is an expiration date on which futures of stocks, indices and commodities expire and/or new futures are issued. On these days, there are often extremely high trading volumes for symbols. Candle volume charts may be overly influenced by them, resulting in no sensible values for these dates. To avoid this issue, ignore these dates.

GENERAL CHART PROPERTIES

When the chart is active, you can edit the chart properties in the toolbox, for example the history length, the axes and legends.

You can also

- set the period and time span, see the chapter Timespans and Periods.
- change the chart look, see the chapter Chart Properties.

CONTENT-DEPENDENT CHART SETTINGS

In the properties of the displayed stock (symbol), you can find additional parameters. Besides the Standard Properties, you can find specific parameters for this chart type.

Candle Volume Properties

Border Color - Here you can select a color for the border.

Show Frame - Here you can select whether the candle frame will be displayed or hidden.

Bullish Color - Here you can select a color and opacity for upward candles.

Bearish Color - Here you can select a color and opacity for downward candles.

EQUI VOLUME CHART

Equivolume charts are a combination of price movement and trading volume charts. The chart gives information on high, low, volume and price direction per trading period.

Equivolume charts were introduced by Richard W. Arms Jr. It was well-known that the trading volume was an important criterion for assessing the current and future situation in the market. Arms consequently introduced the volume - which up to that date was only displayed as an additional histogram - into the chart itself by plotting price versus volume. This way, information about open and close prices and the normal time line are lost.



Equi Volume Chart

TRADING INFORMATION IN EQUI VOLUME CHARTS

The following information is available in the boxes:

- The high per trading period, given by the top line of the box
- The low per trading period, given by the bottom line of the box
- The trading volume, given by the width of the box. Because of this display choice, the timeline information is distorted.
- The direction of the price movement, given by the box color. Colors are usually red or black for falling and green or white for rising prices. The color settings can be changed.

The open and close prices are not given in this chart type. However, this limitation is compensated by the strong visual display of buying powers and selling pressures within a market.

PATTERN IN EQUIVOLUME CHARTS

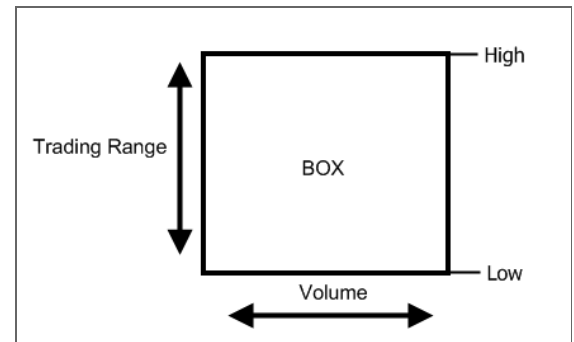
In equivolume charts, four typical patterns can be found that give information on the current market and trends, similar to single candles in candlestick charts. See the graphic for more details.

Narrow Bars - Usually seen in balanced markets. The price movements result from relatively small trading volumes. Several consecutive narrow bars may signal a distinctive trend setting in, though not as strong as one indicated by power bars, see below.

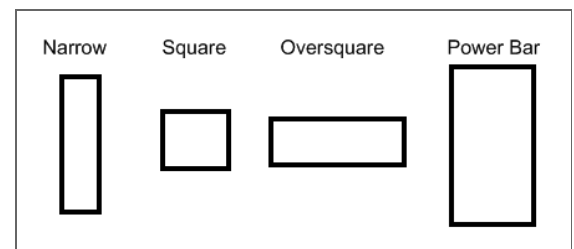
Squares - The price spans are small although there is significant trading activity. The pattern may indicate resistance areas in the chart or appear exactly at the point of trend or resistance lines. Insecurity rules the market, and the resulting wins and losses were achieved under counter pressure.

Oversquares - Both sides of the market are strongly into trading. Large ask volumes have met large bid volumes. The price span is very small. The pattern can be interpreted as trend reversal and be combined with other triggers in charts.

Power Bars - Named after the snack food. In an equivolume chart, Power Bars result from a large price movement based on an active market. They can be interpreted as important breaks and be combined with other triggers in charts.



A Single Bar



Bar Schemata



Equi Volume Pattern

GENERAL CHART PROPERTIES

When the chart is active, you can edit the chart properties in the toolbox, for example the history length, the axes and legends.

You can also

- set the period and time span, see the chapter Timespans and Periods.
- change the chart look, see the chapter Chart Properties.

CONTENT-DEPENDENT CHART SETTINGS

In the properties of the displayed stock (symbol), you can find additional parameters. Besides the Standard Properties, you can find specific parameters for this chart type.

Equi Volume Properties

Border Color - Here you can select a color for the border.

Show Frame - Here you can select whether the candle frame will be displayed or hidden.

Bullish Color - Here you can select a color and opacity for upward boxes.

Bearish Color - Here you can select a color and opacity for downward boxes.

LINKS

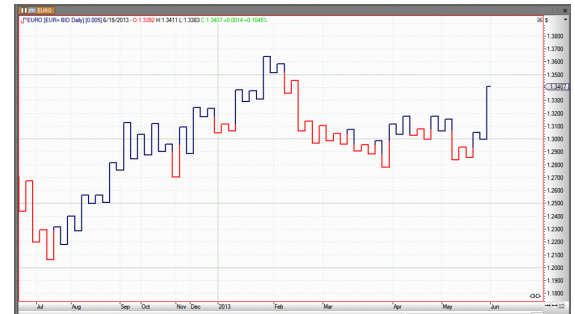
[Equivolume Charts by Richard Arms](#)

KAGI CHART

Kagi charts originated in Japan like the Candlestick Charts. They are related to Renko and Three Line Break charts.

A Kagi chart consists of vertical lines that are connected by horizontal lines. The direction of the vertical lines is drawn depending on the price movements. The time relation, trading volume and chart information such as high and low prices are not taken into account.

Kagi charts display the bid and ask, i.e. the trend of the market.



Kagi Chart

CONSTRUCTION AND INFORMATION IN A KAGI CHART

In a Kagi chart, a vertical line is drawn as long as the current price direction is maintained. If a trend reversal takes place in the price development, the price has to be at least as large as the defined reversal amount to cause a trend reversal in the chart. In case of a trend reversal, a horizontal line is drawn to the next "column" and a vertical line is started in the other price direction.

The time elapsed is unimportant for this kind of charting. The time axis is not linear.

Increasing and decreasing prices are shown in different colors, e.g. black/red. The colors can be set in the properties.

When trading with Kagi charts, remember that the filtering effect inherent to the method leads to delayed signal generation.

GENERAL CHART PROPERTIES

When the chart is active, you can edit the chart properties in the toolbox, for example the history length, the axes and legends.

You can also

- set the period and time span, see the chapter Timespans and Periods.
- change the chart look, see the chapter Chart Properties.

CONTENT-DEPENDENT CHART SETTINGS

In the properties of the displayed stock (symbol), you can find additional parameters. Besides the Standard Properties, you can find specific parameters for this chart type.

Instrument Properties

Reversal - Enter a reversal value here. In case of a trend change, the price has to move by at least the reversal amount before a

new vertical line in the opposite direction is drawn in the Kagi chart.

Kagi Properties

Bullish Color - Here you can select a color for the bullish line.

Bullish Width - Here you can set a bullish line width by entering a number.

Bearish Color - Here you can select a color for the bearish line.

Bearish Width - Here you can set a bearish line width by entering a number.

LINKS AND BOOKS

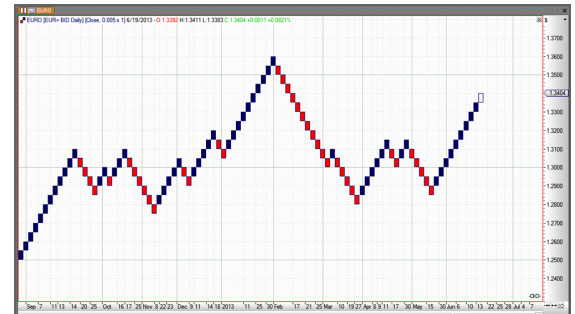
Beyond Candlesticks - Steve Nison

RENKO CHART

Renko charts originated in Japan like the Candlestick Charts. They are related to Kagi and Three Line Break charts. The name derived from the Japanese word *Renga* for bricks.

A Renko chart consists of bricks. Only one input, e.g. the daily close is necessary for drawing a Renko chart. The time relation, trading volume and chart information such as high and low prices are not taken into account.

Renko charts show the trend of the market.



Renko Chart

CONSTRUCTION AND INFORMATION IN A RENKO CHART

In a Renko chart, a brick is drawn if the price moved by a defined amount, called the **bricksiz**e. The price of the current day is always compared to the price of the last brick (which can be much older than a day). The Renko chart drops all price movements smaller than the bricksiz

The time elapsed is unimportant for this kind of charting. The time axis is not linear.

Bricks of downward and upward direction are displayed in different colors, e.g. black/red. The colors can be set in the properties.

When trading with Renko charts, remember that the filtering effect inherent to the method leads to delayed signal generation.

GENERAL CHART PROPERTIES

When the chart is active, you can edit the chart properties in the toolbox, for example the history length, the axes and legends.

You can also

- set the period and time span, see the chapter Timespans and Periods.
- change the chart look, see the chapter Chart Properties.

CONTENT-DEPENDENT CHART SETTINGS

In the properties of the displayed stock (symbol), you can find additional parameters. Besides the Standard Properties, you can find specific parameters for this chart type.

Instrument Properties

Input - Select which data to use for calculating the chart (Open, High, Close, Low, Volume).

Bricksiz - Enter how many ticks give a brick, or enter a percentage value (e.g. "10%"). The bricksiz works like a noise filter. It defines how much the input has to move from the last given brick value before a new brick is drawn.

Reversal - Enter the number of bricks that have to be generated in the reverse direction to cause a change in trend. The higher the value, the stronger the trend reversal has to be to be displayed in the Renko chart.

Compress Bricks - Here you can select whether bricks of the same price will be merged.

Daily Restart - Here you can select whether the Renko chart calculation should start anew each day.

Startvalue from Open - Here you can select whether instead of the input given above, the open value should always be used as start value.

Renko Properties

Color - Here you can select a color for the line border.

Bullish Color - Here you can select a color and opacity for the bullish brick.

Bearish Color - Here you can select a color and opacity for the bearish brick.

Open Color - Select the color for open bricks, i.e. if no close price is available yet.

Show Thresholds - Here you can select whether the price levels that have to be crossed to result in new bricks should be displayed as small, right-aligned lines in the chart.

Artificial Brick Highlight - Here you can select whether bricks that were artificially created due to price gaps between bricks should be highlighted by transparent colors. Select a transparency from the list.

LINKS AND BOOKS

Beyond Candlesticks - Steve Nison

THREE LINE BREAK CHART

Three Line Break charts originated in Japan like the Candlestick Charts. They are related to Kagi and Renko charts.

A Three Line Break chart consists of vertical lines ("boxes") connected to each other. The direction of the lines is drawn based on price movements. The time relation, trading volume and chart information such as high and low prices are not taken into account.

Three Line Break charts are especially useful for detecting changes in trends.



Three Line Break Chart

CONSTRUCTION AND INFORMATION IN A THREE-LINE-BREAK CHART

A Three Line Break chart differs from traditional chart types.

- In principle, a new line is drawn when the price exceeds the previous price's high or low. If the price does not exceed the price benchmarks of the day before, no line is drawn.
- For a "Three Line Break", the same rule applies; however, after three lines of the same color, the price must reverse the amount of the last three lines to cause a trend change in the graph. Price movements that lead neither to a reversal nor to a line of the current direction are ignored.

The number of lines taken into account for the break is given by the **Reversal** parameter. For short-term trades it might be smaller than 3, for long-term trades 4 or higher. The setting "3" is the default value that gave the method its name. Please note that the Reversal parameter gives the number of lines to be taken into account, not a fixed price value as for the related methods Kagi and Renko. Therefore, the price at which the trend change occurs in the chart is relative.

The time elapsed is unimportant for this kind of charting. The time axis is not linear.

Lines (boxes) of downward and upward direction are displayed in different colors, e.g. black/red. The colors can be set in the properties.

For a Three Line Break chart, a typical 'buy' signal is a trend change after three downwards lines, while a 'sell' signal is a trend change after three upwards. Remember that the filtering effect inherent to the method leads to delayed signal generation.

GENERAL CHART PROPERTIES

When the chart is active, you can edit the chart properties in the toolbox, for example the history length, the axes and legends.

You can also

- set the period and time span, see the chapter Timespans and Periods.

- change the chart look, see the chapter Chart Properties.

CONTENT-DEPENDENT CHART SETTINGS

In the properties of the displayed stock (symbol), you can find additional parameters. Besides the Standard Properties, you can find specific parameters for this chart type.

Instrument Properties

Reversal - Enter the number of lines to be taken into account for determining trend reversals.

Input - Select which data to use for calculating the chart (Open, High, Close, Low, Volume).

Three Line Break Properties

Border Color - Here you can select a color for the line border.

Bullish Color - Here you can select a color for the bullish line.

Bearish Color - Here you can select a color for the bearish line.

LINKS AND BOOKS

Beyond Candlesticks - Steve Nison

HEIKIN ASHI CHART

At a first glance Heikin Ashi charts are very similar to Candlestick charts. It is also a charting style originally from Japan and uses candles to visualize prices and their movement. Heikin Ashi charts are based on averaged prices, a result of this is the noticeable absence of gaps between candle bodies which makes it easier to recognize price trends.

Like other candlestick charts it gives information about the open, high, low and close prices and about the direction and magnitude of the price movement. It is this wealth of information and the distinct patterns that makes the candlestick charts so interesting. The direction and movement of prices can be better perceived than by using bars. More complex candlestick patterns are often used as the basis for trading decisions.



A Heikin Ashi Chart

FORMULAS

The Heikin Ashi chart type uses the following formulas to compute each candle's values.

In the following equations let **open**, **close**, **high**, **low** denote the input values and **haOpen**, **haClose**, **haHigh** and **haLow** the corresponding Heikin Ashi values.

$$haOpen_t = \begin{cases} open & t = 1 \\ \frac{haOpen_{t-1} + haClose_{t-1}}{2} & t > 1 \end{cases}$$

$$haClose_t = \frac{open + close + high + low}{4}$$

$$haHigh_t = \text{MAXIMUM}(high, haOpen_t)$$

$$haLow_t = \text{MINIMUM}(low, haOpen_t)$$

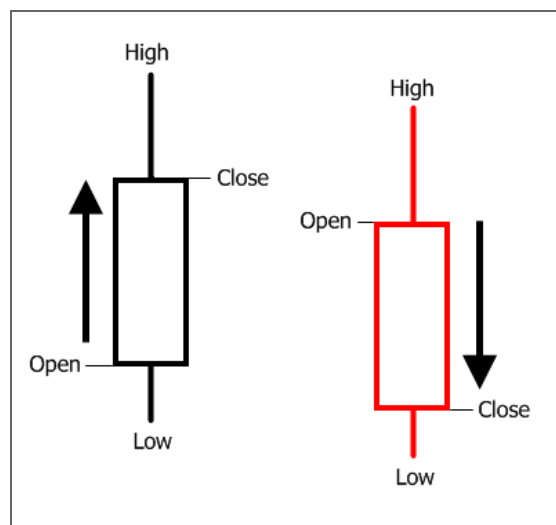
Heikin Ashi formulas

For the first candle Heikin Ashi uses for its **haOpen**-value the **open**-value of the input, subsequent candles use the arithmetic average of the previous candle's **haOpen** and **haClose**-values.

TRADING INFORMATION IN HEIKIN ASHI

The following information is available in the candle body and the shadows (the lines on bottom and top, also called wicks):

- The high of the trading period, given by the end of the upper shadow.
- The low of the trading period, given by the end of the lower shadow.
- The open price, given by the body edge; either the bottom line in the case of an upwards development, or the top line in the case of a downwards development.
- The close price, given by the body edge; either the top line in the case of an upward development, or the bottom line in the case of a downward development.
- The direction of the price movement, given by the body color. Colors are usually red or black for falling and green or white for rising prices. The color settings can be changed.
- The magnitude of the price movement, given by the length of the body and the shadows. The total length of a candlestick between the wicks is the difference between high and low.



A Single Candlestick

More information can be gathered when several candlesticks are seen in context. Patterns of up to three candles are used to take a closer look at the price movements and their quality. For example, candles with bodies of the same color show more pronounced trends. Trading is often done based on a mixture of candlestick pattern analysis and other methods.

GENERAL CHART PROPERTIES

When the chart is active, you can edit the chart properties in the toolbox, for example the history length, the axes and legends.

You can also

- set the trading period and timespans, see the chapter Timespans and Periods.
- change the chart look, see the chapter Chart Properties.

CONTENT-DEPENDENT CHART SETTINGS

In the properties of the displayed stock (symbol), you can find additional parameters. Besides the Standard Properties, you can find specific parameters for this chart type.

Candles Properties

Border Color - Here you can select a color for the border.

Show Frame - Here you can select whether the candle frame will be displayed or hidden.

Bullish Color - Here you can select a color and opacity for upward candles.

Bearish Color - Here you can select a color and opacity for downward candles.

INDICATORS AND STRATEGIES

In the indicator overview, you can find the indicator **Candle Pattern** which includes a long list of typical candlestick patterns to choose from. You can edit these patterns in the graph properties in the toolbox. The set pattern is displayed in the graph.

In the strategies overview, you can find strategies such as **Candle Bearish Engulfing - Exit**, **Candle Big White Candlestick - Exit**, and many more that you can apply to your candlestick chart.

LINKS AND BOOKS

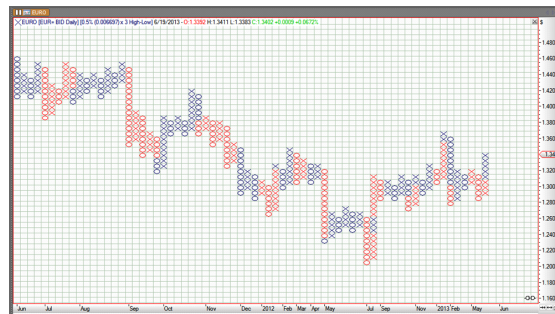
- [Candlestick Charts in the Wikipedia](#)
- [Heikin-Ashi from Dan Vaclu](#)
- [Heikin-Ashi Website](#)

POINT&FIGURE CHART

Point and Figure Charts (P&F charts) are an old charting type that is independent of time.

P&F charts show two kinds of symbols. Depending on the price movement and range, a number of symbols are drawn. The time relation and trading volume are not taken into account.

Point and Figure charts are helpful for trend detection, as they show trends as well as easily identifiable resistance and support zones. There are also Equilla indicators for 45 degree trend lines and support resistance levels.



Point&Figure Chart

TRADING INFORMATION IN A P&F CHART

A P&F chart consists of columns of symbols of the same type.

- Rising prices are drawn with crosses (also called Xs).
- Falling prices are drawn with circles (also called Os).

In a P&F chart these symbols are called boxes and cover a certain price range known as the box size. A new symbol is added to a column once the price movement of the instrument exceeds the box size of the box in trend direction, i.e. in an X-column (rising prices) a new box is drawn when the prices reaches a value above the new box's price range. Movements below that value are ignored. This way, the box size determines how much the price movements are filtered. Large box sizes filter many movements. Small box sizes show movements better, but also have a smaller filter effect.

The reversal parameter of a P&F chart dictates by how many box sizes the price movement has to change in order to establish a new trend which also entails drawing a new column of the opposite type of boxes. For example: for a Reversal value of "3", after a breakout, the counter trend has to exceed three box sizes to change the trend in the chart. Please note that the Reversal parameter gives the number of boxes to be taken into account, not a fixed price value as for the related methods Kagi and Renko. Therefore, the price at which the trend change occurs in the chart is relative. Price movements that lead neither to a reversal nor to a new symbol of the current direction are ignored.

The time elapsed is unimportant for this kind of charting. The time axis is not linear.

Alternating months are displayed in different colors, e.g. black/red. The colors can be set in the properties. (The months are not relevant for the method, but are displayed for a better overview.)

GENERAL CHART PROPERTIES

When the chart is active, you can edit the chart properties in the toolbox, for example the history length, the axes and legends.

You can also

- set the period and time span, see the chapter Timespans and Periods.
- change the chart look, see the chapter Chart Properties.

Select the grid style *Graph Paper* in the Chart Properties if you want the boxes of a P&F chart to be enclosed by grid lines.

CONTENT-DEPENDENT CHART SETTINGS

In the properties of the displayed instrument, you can find additional parameters. Besides the Standard Properties, you can find specific parameters for this chart type.

Instrument Properties

Boxsize Mode - Enter a method to calculate the box size

- *Constant price*
- *Constant points*
- *Percentage of close*, close value on **Base Date** is used
- *Average True Range*, values up to **Base Date** are used
- *Logarithmic*, i.e. box size increases for rising prices, initial box size is based on the open value on **Base Date**

Boxsize - Dependent on the selected **Boxsize Mode** this value is interpreted as

- *Constant price*: a price
- *Constant points*: a number of points
- *Percentage of close*: a percentage value
- *Average True Range*: the period of the ATR, i.e. a weighting factor applied to the EMA
- *Logarithmic*: the percentage by which box sizes increase for rising prices
- For the percentage based values, note that the % sign is implied.

Base Date - This property is used for *Percentage of close*, *Average True Range* and *Logarithmic*. If an invalid date is specified or the date is no longer available, it will be adjusted.

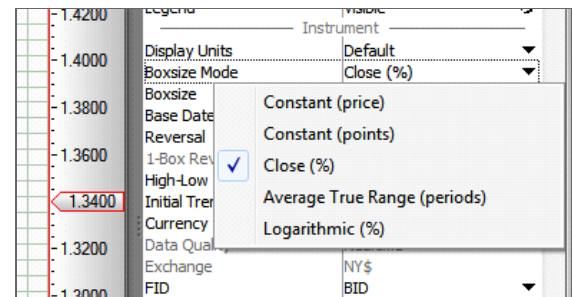
Reversal - Enter the number of boxes to be taken into account for a trend reversal.

1-Box reversal - If the reversal property is set to one, this property determines if one-box reversals are compressed into the next column (Standard option), or if a new column is always created (New Column option).

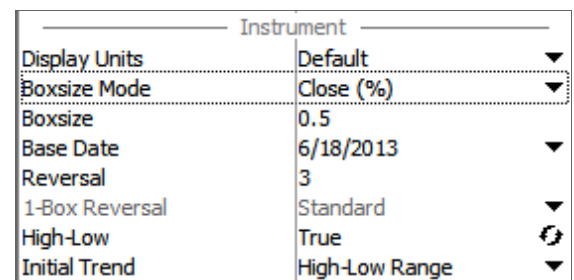
High-Low - Select if boxes and reversals should be calculated against the high and low values instead of the close value.

Initial Trend - Here you can choose how to establish the initial trend. Possible methods are:

- *High-Low Range*: At least **reversal** boxes must be spanned by the highest high and lowest low to establish a trend. If this happens on the first bar, the close value is compared to the midrange of high and low to determine if an X or O is drawn. If



Boxsize modes



Point&Figure Properties

the trend is established on a later bar, the most recently updated extreme value (highest high or lowest low) determines whether the first column contains X or O. This will always draw at least **reversal** boxes in the first column.

- *High-Low Range (simplified)*: This is similar to High-Low Range, but establishes a trend on the first price movement, so it may start a column with just one box. (note: this mode was called *Traditional* in previous versions)
- *Close Range*: This mode also requires **reversal** boxes to be spanned by the highest and lowest value, but only considers close prices. This will always draw at least **reversal** boxes in the first column. (note: this mode was called *Smoothed* in previous versions)
- *Two Box Method*: This method compares the high and low prices of the first bar against high and low prices of subsequent bars. If the high values are found to be at least one box apart, Xs will be drawn. If this is not the case the lows are compared and if they are at least one box apart, Os will be drawn. This will always draw at least two boxes in the first column.

P&F Properties

Color Change - Specifies under which circumstances the colors alternate. This property is available in Tradesignal 7.3 and above.

- *Monthly*: The color changes as soon as a new month starts.
- *Weekly*: The color changes as soon as a new week starts.
- *Daily*: The color changes as soon as a new day starts.
- *Hourly*: The color changes as soon as a new hour starts.
- *Every Column*: The color changes for every column. X columns are drawn using **Color 1** and O columns are drawn using **Color 2**.

Color 1 / Color 2 - Set the symbol colors for alternating months. The displayed symbols toggle between **Color 1** and **Color 2**. In Tradesignal 7.3 and above the **Color Change** property controls how the colors alternate.

INDICATORS ON POINT & FIGURE CHARTS

Indicators and strategies can be applied to a Point & Figure chart like any other chart type. The main issue to be aware of is how the values for Open, High, Low and Close are interpreted. The box value at the top and bottom of a column will determine the high and low respectively. The open will be equal to the low value for an X column and equal to the high for an O column. The close will be equal to the high value for an X column and equal to the low for an O column. An additional field called **Midpoint** is available for input to an indicator, this will be the value of the middle box in a column, or the mean value between the two middle boxes if a column has an even number of boxes.

To apply a Simple Moving Average to a Point & Figure chart using the column Midpoint

1. Drag and drop the indicator **Moving Average Simple** from the Indicator Toolbox onto the point and figure chart.
2. Click on the moving average line in the chart (or the legend), to select the average.
3. In the Property Inspector, locate the **Price** property for the moving average and change it to **Midpoint**.

LINKS AND BOOKS

Point and Figure Charts

Point and Figure Tutorials

The Complete Guide To Point-And-Figure Charting - H. Weber & K. Zieg

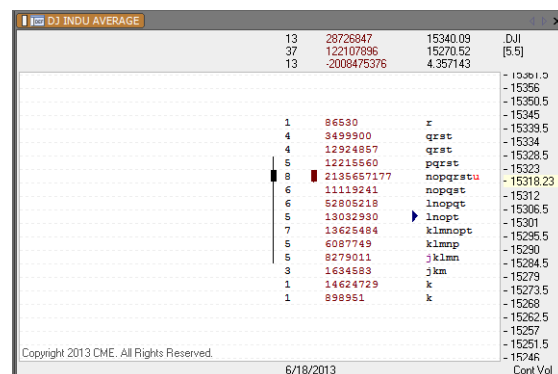
The Definitive Guide to Point an Figure - J. Du Plessis

The Essential Application for Forecasting and Tracking Market Prices - T.J. Dorsey

MARKET PROFILE

Market Profile charts were invented in the 1980's. The goal was to offer a better analysis method to traders at the US commodity markets. The development was done in cooperation between the trader Peter Steidlmayer and the Chicago Board of Trade (CBOT).

Market Profile charts are different from the normal display of the price movements versus the time line. While the prices are still given on the x-axis, the time line is given in units of "Time Price Opportunities" (TPO) that are marked with letters. This results in a more or less bell-shaped distribution of the TPOs over the price span of the trading day.



Market Profile Chart

Use the Market Profile chart for

- Analyzing the distribution of trading activity
- Analyzing the "Fair Value" span with most of the trading activity
- Analyzing the volume distribution over the trading day
- Analyzing of trend movements

Tradesignal offers you Market Profile charts as a designated chart type. However, it is not available for all versions. As an alternative for displaying the volume/price correlation, you can use the indicator "Price Volume Profile" that is delivered with Tradesignal.

Due to the special focus of the tick-based Market Profile, it is usually only displayed for one or two trading days. If you want to display more than two trading days, the history length has to be set to the maximal value of 500000 ticks. If your data provider offers this history length, you can display up to eight days as Market Profile.

CREATING A MARKET PROFILE

For a Market Profile, stocks, futures, commodities, currencies etc. can be used as long as they show a high trading frequency per day.

For a quick look at the chart type, right-click on an index in the symbol list and select **Open in Market Profile**.

CONSTRUCTION AND INFORMATION IN A MARKET PROFILE

The time units for the graph are fixed intervals called "Time Price Opportunities" (TPOs). They are basically trading periods. The standard setting is 30 minutes but can also be set to one hour.

For each TPO, a letter is used, starting with a capital "A". When the start of the trading is entered correctly, the "A" is displayed in the chart for the first price in the first TPO. For each new price level traded in that TPO, the letter "A" is added. The size of each price level is set in the **Scale factor**. For example, for a scale factor of "2", the price has to change 2 points (or cents) to open a

new price level.

After the first TPO, the price levels traded in the second TPO are displayed in the chart with a "B", in the third with "C" and so forth.

The result of this continuous recording is a pictorial representation of the trading activity in relation to the time line. In the chart given as example below, the price had a very small range for a while until it broke out. The large protrusion for the letters D and E shows that in the range of 6370 points, an equilibrium was formed. Such an equilibrium is called the *Fair Value Area*. At the end of the trading day, the position of the *Fair Value Areas* in regard to their position in the total price range can be evaluated.

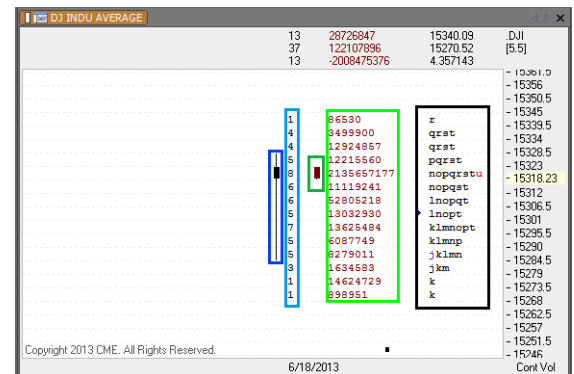
As additional information, a brown and a black bar are displayed. The black bar spans 70% of the time with the highest trading activity. The brown bar spans 70% of the highest trading activity (in volume). The two rectangles mark the respective maxima.

ELEMENTS IN A MARKET PROFILE CHART

BLACK FRAMED AREA IN THE EXAMPLE GRAPH

Here, the price versus time is displayed. Each letter stands for a TPO (Time Price Opportunity). Above this area, in the header of the chart, the following data is displayed (from top to bottom):

1. The highest price level traded
2. The lowest price level traded
3. The ratio of number of letters and the high/low span



LIGHT-GREEN FRAMED AREA IN THE EXAMPLE GRAPH

Here you can find the number of traded contracts (contract volume) or the tick volume for each price level. In the example, 1302 contracts were traded (bought and sold) in the price span 6381-6382. The displayed volume is specified with the parameter **Volume Type** in the properties.

DARK-GREEN FRAMED AREA IN THE EXAMPLE GRAPH

Here you can find a brown bar that marks the range of 70 % of volume activity. The small rectangle marks the price level with the highest volume activity. This point is called *Volume Point of Control*. Above this area, in the header of the chart, the following data is displayed (from top to bottom):

1. Volumes traded above the Point of Control
2. Volumes traded below the Point of Control
3. Total trading volume

LIGHT-BLUE FRAMED AREA IN THE EXAMPLE GRAPH

Here you can find the sum of TPOs (periods) in which the respective price levels were traded. For example, the range 6376-6377 was traded in a total of six periods.

DARK-BLUE FRAMED AREA IN THE EXAMPLE GRAPH

Here you can find the black bar that marks the range of 70% activity in the time line. The small rectangle marks the price level with the highest trading activity. This point is called *TPO Point of Control*.

MARKET PROFILE CHART PROPERTIES

Here you can find the most important parameters for this chart type.

History Length - The Market Profile is created on a tick basis. Enter the number of ticks to be loaded into the Market Profile. The history length determines how many days will be displayed as Market Profile.

Interval - This is the interval that each TPO represents in the chart. The default value is 30 minutes but can also be set to 1 hour.

Scale Factor - Enter the price span units. When the price moves by this span during a TPO, the current letter is drawn at that price level. For example, if the scale factor is "2" and the current price is at "6370" in the TPO "C", a new "C" is drawn in the line above the original one if the price moves up to 6372.

Show Gaps - Select here whether gaps in the trading (TPOs without trades) are to be displayed.

Initial Balance - Enter the number of TPOs that should be part of the start period. TPOs in the start period are labeled in a special color.

Start Time - Here the time for the market start has to be entered, i.e. "9:30" for stock trading. This is important for a correct display of the Market Profile.

Volume Type - Select here whether the volume should be displayed as **Contract Volume** (number of traded contracts) or as **Tick Volume** (number of ticks = trades for this price span; necessary for currencies, for example). The traded volumes are displayed to the left of the letters (light-green area in the example chart).

Zoom Factor - Here you can enter an optical zoom factor. The higher the number, the more the profile is magnified.

LINKS

Tradesignal Online Lexikon

DRAWING TOOLS

TEXT

With the text tool, you can enter notes or captions into a chart, e.g. before publishing it in the Tradesignal web forum.

You can find the text tool in the *Chart Tools* group in the toolbar.

ENTERING TEXT

1. Open a symbol chart.
2. Click on the text tool in the toolbar.
3. Click into the chart. A text field with four anchor points appears.

Now you can enter the text.



Text

EDITING TEXT

- To edit the text, click on it with the mouse (so that it is selected) and then click once again, so that the text field is highlighted. Now you can edit the text.
- To move the text, drag it with the mouse (the mouse pointer turns into a hand symbol) and move it freely. Release the mouse button when done.
- You can duplicate the text by double-clicking it.
- When the text is selected, you can edit the text properties in the toolbox.
- To delete the text, right-click on it and select **Delete** from the context menu. Alternatively, select the text and press the **DEL** key.

PROPERTIES

Font - Here you can change the font type and size, e.g. to "Arial bold 10".

Color - Here you can change the text color, e.g. to red.

Background - Here you can set a background color and opacity for the text object, e.g. black.

Frame - Here you can select whether you want to display a frame (in the same color as the text) around the text.

Anchor - Here you can select how the text is placed in relation to the chart data. Choose an option from the drop-down list.

TREND LINE

The simplest tool for a chart analysis is the trend line. With it, trends in price movements can be visualized and emphasized. Trading decisions may be based on analysis of trend lines. Trend lines may be used for:

- Visualizing trends
- Marking market phases
- Marking resistance and support levels

You can find the trend line tool in the *Chart Tools* group in the toolbar.



Trend Line

ENTERING A TREND LINE

1. Open a symbol chart.
2. Click on the trend line tool in the toolbar.
3. Click into the chart to set the starting point of the trend line.
4. Keep the mouse button pressed and move the pointer to the end point of the trend line. Release the mouse button.

You can align the anchor points of the trend line to a certain price level. To do so, activate the Snap Mode by clicking on the magnet symbol in the toolbar. Now when you draw a trend line in the chart, it is positioned exactly at high and/or low points.

EDITING A TREND LINE

- To move the trend line, drag it with the mouse (the mouse pointer turns into a hand symbol) and move it freely. Release the mouse button when done.
- To change the trend line angle, click the anchor points of the line (the mouse pointer turns into an axis of coordinates symbol) and drag into another position.
- Note that the trend line cannot be moved freely if the Snap Mode is active. If necessary, switch it off by clicking the magnet symbol in the toolbar.
- You can duplicate a trend line in three ways (see the chapter Chart Tools).
- To extend the length of the trend line until the right or left edge of the chart, double-click on the respective anchor point at the end of the trend line.
- When the trend line is selected, you can edit the properties in the toolbox.
- The alert function is available for the trend line. To set an alert, set the **Alert Mode** from *Disabled* to one of the options (see below).
- To delete the trend line, right-click on it and select **Delete** from the context menu. Alternatively, select the trend line and press the **DEL** key.

USING THE MEASUREMENT FUNCTION

In the trend line properties, you can switch on the measurement function.

The following information is displayed for the time axis:

- Number of bars (trading periods) between the two anchor points
- Number of calendar days between the two anchor points

Example: "47 Bars (2d)"

The following information is displayed for the price axis:

- Absolute price difference between the price levels of the anchor points
- Relative price movement between the price levels of the anchor points
- Relative gradient between periods for the slope between the two anchor points

Example:

"3.48 (+5.7)"

"0.07/Bar"

PROPERTIES

TREND LINE PROPERTIES

Extend Left/Extend Right - Select whether the line is to be extended to the left and/or right until the edge of the chart.

GRAPHIC PROPERTIES

The following properties can be set for the trend line tool:

Color - Here you can select a color for the line.

Width - Here you can set a line width by entering a number.

Style - Here you can select a line style, e.g. solid or dotted.

Shadow - Here you can set a shadow for emphasizing the line with a 3D effect. Four settings are available.

Label - Here you can set the value label for the last data point to visible or hidden. The label is only displayed when the trend line is drawn through, below or above the last data point (candle, bar etc.). In Tradesignal you can also specify the location at which the label should be displayed.

Measurement - Here you can switch the measurement function on or off.

ALERT PROPERTIES

The trend line tool can be used with various kinds of alerts.

Alert Direction - Here you can select from the list whether crossings of all types lead to alerts or only crossings in a certain direction.

Alert Mode - Here you can select an alert mode from the list. Select whether every crossing is considered or only one crossing per trading period or trend line.

Alert Name - Here you can specify a name for alerts generated by this tool. It will be included in the signal field of the alert.

Alert Trigger - Here you can select when an alert should be triggered.

- *Crossed*: Triggers when the price crosses the tool, e.g. goes from below the tool to above the tool
- *Hit or Crossed*: Same as above, but also triggers when the price merely hits the tool. This setting is useful for a Point&Figure Chart.

For further information on alert modes, see the chapter Alerts.

TREND CHANNEL

The trend channel is a standard tool in chart analysis. It offers two parallel lines that are helpful for visualizing trends. Trading decisions may be based on the trend channel's alert function.

Trend channels may be used for visualizing:

- strong price movements
- resistance and support levels
- chart patterns
- sideways markets



Trend Channel

You can find the trend channel tool in the *Chart Tools* group in the toolbar.

ENTERING A TREND CHANNEL

1. Open a symbol chart.
2. Click on the trend channel tool in the toolbar.
3. Click into the chart to set the starting point of the trend channel.
4. Keep the mouse button pressed and move the pointer to the end point of the trend channel. Release the mouse button.

You can align the anchor points of the trend channel to a certain price level. To do so, activate the Snap Mode by clicking on the magnet symbol in the toolbar. Now when you draw a trend channel in the chart, the middle line is positioned exactly at high and/or low points.

EDITING A TREND CHANNEL

- To move the trend channel, drag it with the mouse (the mouse pointer turns into a hand symbol) and move it freely. Release the mouse button when done.
- To move the lines of the trend channels up or down, click the center point of a line (the mouse pointer changes into a double arrow) and drag it.
- To change the trend channel angle, click the anchor points of the channel (the mouse pointer turns into an x-y axis symbol) and drag into another position.

Note that the trend channel cannot be moved freely if the Snap Mode is active. If necessary, switch it off by clicking the magnet symbol in the toolbar.

- You can duplicate a trend channel in three ways (see the chapter Chart Tools).
- To extend the length of the trend channel until the right or left edge of the chart, double-click on anchor points at the right or left end of the channel.
- When the trend channel is selected, you can edit the properties in the toolbox.
- The alert function is available for the trend channel. To set an alert, set the **Alert Mode** from *Disabled* to one of the options

(see below).

- To delete the trend channel, right-click on it and select **Delete** from the context menu. Alternatively, select the trend channel and press the **DEL** key.

PROPERTIES

TREND CHANNEL PROPERTIES

Distance - Here you can enter a distance between the two lines of the trend channel. The value changes if you move the lines independently. You can also directly enter the value here.

Always parallel - Select whether the lines of the trend channel are always be kept parallel. (If you change the chart display from linear to logarithmic, the drawing elements in the chart may "bend" and change their angles. In the case of trend channels, this may lead to non-parallel lines. Set the option to *true* to avoid this distortion.)

Extend Left/Extend Right - Select whether the lines are to be extended to the left and/or right until the edge of the chart.

GRAPHIC PROPERTIES

The following properties can be set for the trend line tool:

Color - Here you can select a color for the lines.

Width - Here you can set a line width by entering a number.

Style - Here you can select a line style, e.g. solid or dotted.

Shadow - Here you can set a shadow for emphasizing the lines with a 3D effect. Four settings are available.

Fill Color - Here you can select a color and opacity for the channel area.

Label - Here you can set the value label for the last data point to visible or hidden. The label is only displayed when the trend line is drawn through, below or above the last data point (candle, bar etc.). In Tradesignal you can also specify the location at which the label should be displayed.

ALERT PROPERTIES

The trend channel tool can be used with various kinds of alerts.

Alert Direction - Here you can select from the list whether crossings of all types lead to alerts or only crossings in a certain direction.

Alert Mode - Here you can select an alert mode from the list. Select whether every crossing is considered or only one crossing per trading period or trend line.

Alert Name - Here you can specify a name for alerts generated by this tool. It will be included in the signal field of the alert.

Alert Trigger - Here you can select when an alert should be triggered.

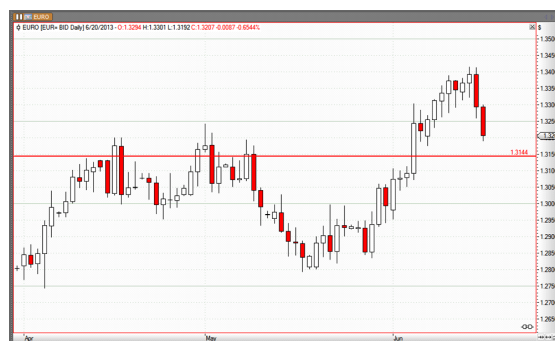
- *Crossed*: Triggers when the price crosses the tool, e.g. goes from below the tool to above the tool
- *Hit or Crossed*: Same as above, but also triggers when the price merely hits the tool. This setting is useful for a Point&Figure Chart.
- *Break-in/Break-out*: Treats the tool not as two separate lines but as a channel instead and triggers an alert, when the price breaks into or out of the tool's channel.

For further information on alert modes, see the chapter Alerts.

STOP LINE

The stop line is a simple horizontal line that is defined by only the value on the price axis. Entered for one level, the line is extended to both sides of the chart. The stop line has an alert function and can be used for precise timing of buy and sell orders. The stop line may be used for

- drawing horizontal lines
- marking resistance and support levels
- exactly measuring price movements
- placing a line at a user-defined price level



Stop Line

ENTERING A STOP LINE

1. Open a symbol chart.
2. Click on the stop line tool in the toolbar.
3. Click into the chart. A horizontal stop line is drawn. The current value is given at the left side of the chart.

You can align the anchor points of the stop line to a certain price level. To do so, activate the Snap Mode by clicking on the magnet symbol in the toolbar. Now when you enter a stop line in the chart, it is positioned exactly at a high or low point.

EDITING A STOP LINE

- To move the stop line, drag it with the mouse (the mouse pointer turns into a double arrow) and move it up or down. Release the mouse button when done.

Note that the stop line cannot be moved freely if the Snap Mode is active. If necessary, switch it off by clicking the magnet symbol in the toolbar.

- You can duplicate the stop line by double-clicking it.
- When the stop line is selected, you can edit the properties in the toolbox.
- For the stop line, the alert function is available. To set an alert, set the **Alert Mode** in the properties in the toolbox from *Disabled* to one of the options, see below.
- To delete the stop line, right-click on it and select **Delete** from the context menu. Alternatively, select the stop line and press the **DEL** key.

PROPERTIES

GRAPHIC PROPERTIES

The following properties can be set for the trend line tool:

Color - Here you can select a color for the line.

Width - Here you can set a line width by entering a number.

Style - Here you can select a line style, e.g. solid or dotted.

Label Alignment - Here you can set the value label to visible or hidden and align it to the stop line.

ALERT PROPERTIES

The stop line tool can be used with various kinds of alerts.

Alert Direction - Here you can select from the list, if crossings of all kinds lead to alerts or only crossings of a certain direction.

Alert Mode - Here you can select an alert mode from the list. Select if every crossing is considered or only one crossing per trading period or trend line.

Alert Name - Here you can specify a name for alerts generated by this tool. It will be included in the signal field of the alert.

Alert Trigger - Here you can select when an alert should be triggered.

- *Crossed*: Triggers when the price crosses the tool, e.g. goes from below the tool to above the tool
- *Hit or Crossed*: Same as above, but also triggers when the price merely hits the tool. This setting is useful for a Point&Figure Chart.

For further information on alert modes, see the chapter Alerts.

ANDREW'S PITCHFORK

Andrew's Pitchfork (originally called "median line studies") is a drawing tool which, if correctly aligned, is also a geometric indicator. Andrew's Pitchfork consists of three parallel lines and is especially useful for finding resistance and support levels.

You can find the tool in the *Chart Tools* group in the toolbar.

ENTERING ANDREW'S PITCHFORK



Andrew's Pitchfork

Three points are necessary to enter a Andrew's Pitchfork.

- The start (pivot) point of a trend (high or low). This is the starting point of the handle.
- The two corner points of a major correction against the trend. These will be the starting points of the prongs.

Proceed as follows:

1. Open a symbol chart.
2. Click on the Andrew's Pitchfork tool in the toolbar.
3. Click into the chart on one of the corners of the correction movement.
4. Keep the mouse button pressed and move the pointer to the second corner of the correction movement. Release the mouse button.
5. Drag the start point (anchor point of the handle) to the correct position by moving it to the start point of the trend.

The resulting pitchfork consists of a handle with a middle line and two prongs. All lines can serve as resistance and support levels in price movements.

EDITING ANDREW'S PITCHFORK

- To move Andrew's Pitchfork, drag it with the mouse (the mouse pointer turns into a hand symbol) and move it freely. Release the mouse button when done.
- To change the direction of Andrew's Pitchfork, click the pivot point of the handle (the mouse pointer changes into an x-y axis symbol) and move it. This will make the fork rotate.
- You can edit the corner points of the prongs. Click these anchor points (the mouse pointer turns into an x-y axis symbol) and drag them to another position. Due to the line coupling, the fork's appearance changes.
- You can duplicate Andrew's Pitchfork by double-clicking on it.
- When Andrew's Pitchfork is selected, you can edit the properties in the toolbox.
- The alert function is available for Andrew's Pitchfork. To set an alert, set the **Alert Mode** from *Disabled* to one of the options (see below).
- To delete Andrew's Pitchfork, right-click on it and select **Delete** from the context menu. Alternatively, select Andrew's

Pitchfork and press the **DEL** key.

PROPERTIES

GRAPHIC PROPERTIES

The following properties can be set for the Andrew's Pitchfork tool:

Handle Color - Here you can select a color for the handle line.

Prongs Color - Here you can select a color for the prong lines.

Width - Here you can set a width for all lines by entering a number.

Handle Style - Here you can select a line style for the handle, e.g. solid or dotted.

Prongs Style - Here you can select a line style for the prongs, e.g. solid or dotted.

Shadow - Here you can set a shadow for emphasizing all lines with a 3D effect. Four settings are available.

Base - Here you can select whether the base line (the line connecting the two anchor points of the prongs) is to be visible or hidden.

Label - Here you can set the value label for the current data point to visible or hidden. In Tradesignal you can also specify the location at which the label should be displayed

ALERT PROPERTIES

The Andrew's Pitchfork tool can be used with various kinds of alerts.

Alert Direction - Here you can select from the list whether crossings of all types lead to alerts or only crossings in a certain direction.

Alert Mode - Here you can select an alert mode from the list. Select whether every crossing is considered or only one crossing per trading period or trend line.

Alert Name - Here you can specify a name for alerts generated by this tool. It will be included in the signal field of the alert.

Alert Trigger - Here you can select when an alert should be triggered.

- *Crossed*: Triggers when the price crosses the tool, e.g. goes from below the tool to above the tool
- *Hit or Crossed*: Same as above, but also triggers when the price merely hits the tool. This setting is useful for a Point&Figure Chart.

- *Break-in/Break-out*. Treats the tool not as two separate lines but as a channel instead and triggers an alert, when the price breaks into or out of the tool's channel.

For further information on alert modes, see the chapter Alerts.

LINKS AND BOOKS

<http://www.investopedia.com/articles/forex/05/AndrewsPitchfork.asp>

LINEAR REGRESSION CHANNEL

The linear regression channel is a drawing tool with a calculation function. You as the user define the start and end point of the channel as well as the calculation method for the outer lines. The definite position of the channel in the chart, however, results from the calculation of the mid-line.

The mid-line of the channel is calculated on the basis of linear regression with the least square method. In this method, the line that is drawn through a number of data points effectively reflects the trend. The channel is constructed with lines above and below the mid-line, in defined distances.



Linear Regression Channel

The linear regression channel may be used for:

- visualizing trends
- searching swing points
- finding resistance and support levels

You can find the tool in the *Chart Tools* group in the toolbar.

ENTERING A LINEAR REGRESSION CHANNEL

1. Open a symbol chart.
2. Click on the linear regression channel tool in the toolbar.
3. Click into the chart to set the starting point of the linear regression channel.
4. Keep the mouse button pressed and move the pointer to the end point of the linear regression channel. Release the mouse button.

By default, the linear regression channel is drawn with a standard deviation of "2" from the linear regression line. Therefore, the width of the channel depends on the price volatility over the time span between the channel start and end.

Since the linear regression channel is based on calculation, it aligns to the data points almost as if in Snap Mode.

EDITING A LINEAR REGRESSION CHANNEL

- To move the linear regression channel, drag it with the mouse (the mouse pointer turns into a hand symbol) and move it. Release the mouse button when done. Note that since the linear regression channel is based on calculation, it aligns to the data points almost as if in Snap Mode.
- To extend the length of the linear regression channel, i.e. the data points used for calculation, click an anchor point at the end of the middle line (the mouse pointer changes into a double arrow) and drag it.
- To extend the length of the channel until the right or left edge of the chart, double-click on the respective anchor point at the

end of the middle line.

- You can duplicate a linear regression channel by double-clicking on it.
- When the linear regression channel is selected, you can edit the properties in the toolbox.
- You can edit the width of the linear regression channel either by changing the **StdDev** parameters for the Standard Deviation or by selecting another calculation method for the channel (see below).
- The alert function is available for the linear regression channel. To set an alert, set the **Alert Mode** from *Disabled* to one of the options (see below).
- To delete the linear regression channel, right-click on it and select **Delete** from the context menu. Alternatively, select the linear regression channel and press the **DEL** key.

PROPERTIES

REGRESSION CHANNEL PROPERTIES

Extend Left/Extend Right - Select if the channel is to be extended to the left and/or right until the edge of the chart.

Channel Method - Select the method that will be used for calculating the channel:

- **Standard Deviation** - The width of the channel is given by the measured standard deviation in the channel's time span. The **StdDvs up** and **StdDvs down** parameters give the offset (above and below) from the regression line.
- **High / Low** - The channel includes the highest and lowest price within the time span. This channel method is similar to the classic Trend Channel.
- **Percentage** - The width is given as percentage of the price at the last anchor point. The **%** parameter gives the offset from the regression line.

GRAPHIC PROPERTIES

The following properties can be set for the linear regression tool:

Color - Here you can select a color for the lines.

Width - Here you can set a line width by entering a number.

Style - Here you can select a line style, e.g. solid or dotted.

Shadow - Here you can set a shadow for emphasizing the lines with a 3D effect. Four settings are available.

Fill Color - Here you can select a color and opacity for the channel area.

Label - Here you can set the value label for the current data point to visible or hidden. In Tradesignal you can also specify the location at which the label should be displayed.

ALERT PROPERTIES

The linear regression channel tool can be used with various kinds of alerts.

Alert Direction - Here you can select from the list whether crossings of all types lead to alerts or only crossings in a certain direction.

Alert Mode - Here you can select an alert mode from the list. Select whether every crossing is considered or only one crossing per trading period or trend line.

Alert Name - Here you can specify a name for alerts generated by this tool. It will be included in the signal field of the alert.

Alert Trigger - Here you can select when an alert should be triggered.

- *Crossed*: Triggers when the price crosses the tool, e.g. goes from below the tool to above the tool
- *Hit or Crossed*: Same as above, but also triggers when the price merely hits the tool. This setting is useful for a Point&Figure Chart.
- *Break-in/Break-out*: Treats the tool not as two separate lines but as a channel instead and triggers an alert, when the price breaks into or out of the tool's channel.

For further information on alert modes, see the chapter Alerts.

LINKS AND BOOKS

"Trading the Regression Channel" - Gilbert Raff

GANN LINE

The Gann line is a drawing tool based on the analysis methods of W.D. Gann. In his publications around 1929 Gann presented methods based on the premise that the market obeys mathematical - geometrical - laws.

He proposed a relation between the price dimension and time dimension, which construct a price pattern. Thus the idea of constructing trend lines based on angles and time was born.

You can use Gann lines for

- Marking trends in price movements
- Searching support and resistance price levels

You can find the Gann line in the *Chart Tools* group in the toolbar.



Gann Line

ENTERING A GANN LINE

1. Open a symbol chart.
2. Click on the Gann line tool in the toolbar.
3. Click into the chart to set the starting point.

The Gann line is drawn. The default setting of the Gann line is an angle of 45°. This complies to a 1:1 ratio of time and price. The angle results from the property **Runs** (**Rises** is a calculated value). Change the **Runs** value to change the angles. With a **Runs** value of 2, a 1:2 ratio results.

EDITING A GANN LINE

- To move the Gann line, drag the line, i.e. the anchor point, with the mouse (the mouse pointer turns into a hand symbol) and move it freely. Release the mouse button when done.
- You can duplicate a Gann line by double-clicking on it.
- When the Gann line is selected, you can edit the properties in the toolbox.
- To delete the Gann line, right-click on it and select **Delete** from the context menu. Alternatively, select the Gann line and press the **DEL** key.

PROPERTIES

Runs - Here you can enter how many candles/bars a price on the price axis should represent.

Rises - This value is calculated from the **Runs** value and the line properties when first drawing the Gann line. Will not be adapted

when you move or edit the Gann line.

Color - Here you can edit the line color, e.g. to red.

Width - Here you change the thickness of the guide line in pixels, e.g. to "3".

Style - Here you can edit the line style, e.g. set it to "Dot".

LINKS

[W.D.Gann Investopedia](#)

ARROW

Like the guide line and other tools, the arrow helps you to illustrate special situations in charts. This, in addition with analysis tools, allows you to create more meaningful charts.

You can find the arrow tool in the *Chart Tools* group in the toolbar.

ENTERING AN ARROW

1. Open a symbol chart.
2. Click on the arrow tool in the toolbar.
3. Click into the chart to set the starting point.
4. Keep the mouse button pressed while moving the pointer to the end point, and release it there.

At the start point, a text box appears. It includes general chart information and analysis results (open, high, low...) for the data encompassed by the arrow.

You can align the anchor points of the arrow to a certain price level. To do so, activate the Snap Mode by clicking on the magnet symbol in the toolbar. Now when you draw an arrow in the chart, it is positioned exactly at high and/or low points.

EDITING AN ARROW

- To move the arrow, drag it with the mouse (the mouse pointer turns into a hand symbol) and move it freely. Release the mouse button when done. Note that although the text box will move with the arrow, the text box contents will not be refreshed. This might mislead you in interpretation.
- To change the arrow angle, click an anchor point of the arrow (the mouse pointer turns into an x-y axis symbol) and drag it to another position.

Note that the arrow cannot be moved freely if the Snap Mode is active. If necessary, switch it off by clicking the magnet symbol in the toolbar.

- You can duplicate an arrow by double-clicking on it.
- When the arrow is selected, you can edit the properties in the toolbox.
- To delete the arrow, right-click on it and select **Delete** from the context menu. Alternatively, select the trend line and press the **DEL** key.

PROPERTIES

User Text - Here you can enter a title for the text box, which will be displayed as its header line.



Arrow

Width - Here you can enter the thickness of the arrow in pixel, e.g. "3".

Arrowhead Angle - Here you can set how narrow (small number) or wide (high number) the arrowhead will be.

Arrowhead Size - Here you can set the length of the arrowhead.

Step Mode - Here you can select whether the line between the start and end point is to be straight or stepped (bent shortly below the end point).

Arrowhead Shape - Here you can select between four different shapes, e.g. "triangular".

Style - Here you can select the line style of the arrow, e.g. "Dot".

Arrow Border Color - Here you can edit the arrow color.

Arrow Fill Color - Here you can edit the fill color for the arrowhead.

Autotext - Here you can set how many details are displayed in the text box. The box can include basic chart information or also indicator data etc.

Autotext Fill Color - Here you can edit the background color for the text box.

Autotext Color - Here you can edit the text color in the text box.

Guide Line - Select if the guide line should always be visible. Otherwise it will be temporarily displayed to help you setting the exact arrow's end point.

GUIDE LINE

The guide line is a tool for marking events in a trading period. With it, you can easily emphasize a candle, a bar or the signal of an indicator. In Tradesignal, the guide line comes with a date label.

You can find the guide line in the *Chart Tools* group in the toolbar.



Guide Line

ENTERING A GUIDE LINE

1. Open a symbol chart.
2. Click on the guide line tool in the toolbar.
3. Click into the chart to set the anchor point of the guide line.

The guide line is drawn into the graph. The default setting of the tool is that a guide line drawn in a chart will be displayed in all subcharts. However, you can move or delete this guide line only in the chart in which it was originally drawn, because only there can you take hold of it with the mouse.

You can align the guide line to a certain price level. To do so, activate the Snap Mode by clicking on the magnet symbol in the toolbar. Now when you draw the guide line in the chart, it is positioned exactly at high and/or low points.

EDITING A GUIDE LINE

- To move the guide line, drag it with the mouse (the mouse pointer turns into a vertical double arrow symbol) and move it freely. Release the mouse button when done.

Note that the guide line cannot be moved freely if the Snap Mode is active. If necessary, switch it off by clicking the magnet symbol in the toolbar.

- You can duplicate a guide line by double-clicking on it.
- When the guide line is selected, you can edit the properties in the toolbox.
- To delete the guide line, right-click on it and select **Delete** from the context menu. Alternatively, select the trend line and press the **DEL** key.

PROPERTIES

Color - Here you can edit the line color, e.g. to red.

Style - Here you can edit the line style, e.g. set it to "Dot".

Width - Here you change the thickness of the guide line in pixel, e.g. to "3".

Scope - Here you can select whether the guide line is displayed only in the graph in which it was drawn or in all of them (default).

Label - Here you can select whether the data point label with date and time (for shorter periods) is visible or hidden in the chart. In Tradesignal you can also specify the the text orientation.

Label Alignment - Here you can select at which location the label should be displayed.

ELLIPSE

The ellipse is a simple drawing object. Typically ellipses are used for highlighting the expansion of a consolidation zone, based on the idea of a harmonic price movement.

You can find the ellipse tool in the *Chart Tools* group in the toolbar.

ENTERING AN ELLIPSE

1. Open a symbol chart.
2. Click on the ellipse line tool in the toolbar.
3. Click into the chart to set the anchor point.
4. Keep the mouse button pressed and draw the ellipse. Release the mouse button when done.



Ellipse

EDITING AN ELLIPSE

- To move the ellipse, drag it with the mouse (the mouse pointer turns into a hand symbol) and move it freely. Release the mouse button when done.
- You can duplicate an ellipse by double-clicking on it.
- When the ellipse is selected, you can edit the properties in the toolbox.
- To delete the ellipse, right-click on it and select **Delete** from the context menu. Alternatively, select the ellipse and press the **DEL** key.

PROPERTIES

Fill Color - Here you can set up which color and opacity you want to use for the ellipse area. The higher the opacity, the lower the transparency, i.e. the less you can see the original chart behind it.

Border Color - Here you can set the border color of the ellipse.

RECTANGLE

The rectangle is a simple drawing tool. Typically rectangles are used for highlighting mean-reverting markets, support and resistance price levels or chart patterns.

You can find the rectangle tool in the *Chart Tools* group in the toolbar.

ENTERING A RECTANGLE

1. Open a symbol chart.
2. Click on the rectangle tool in the toolbar.
3. Click into the chart to set the anchor point.
4. Keep the mouse button pressed and draw the rectangle. Release the mouse button when done.



Rectangle

EDITING A RECTANGLE

- To move the rectangle, drag it with the mouse (the mouse pointer turns into a hand symbol) and move it freely. Release the mouse button when done.
- To move single sides of the rectangle up/down or left/right, click an anchor point in the middle of a side and drag it.
- To change the size of the rectangle, click an anchor point in the corner of the rectangle and drag it.
- You can duplicate a rectangle by double-clicking on it.
- When the rectangle is selected, you can edit the properties in the toolbox.
- To delete the rectangle, right-click on it and select **Delete** from the context menu. Alternatively, select the rectangle and press the **DEL** key.

PROPERTIES

Fill Color - Here you can set up which color and opacity you want to use for the rectangle area. The higher the opacity, the lower the transparency, i.e. the less you can see the original chart behind it.

Border Color - Here you can set the border color of the rectangle.

CIRCLE

The circle is a simple drawing object. Typically circles are used for highlighting intersections of trend lines, chart patterns or target prices.

You can find the circle tool in the *Chart Tools* group in the toolbar.

ENTERING A CIRCLE

1. Open a symbol chart.
2. Click on the circle tool in the toolbar.
3. Click into the chart to set the anchor point.
4. Keep the mouse button pressed and draw the circle. Release the mouse button when done.



Circle

EDITING A CIRCLE

- To move the circle, drag it with the mouse (the mouse pointer turns into a hand symbol) and move it freely. Release the mouse button when done.
- You can duplicate a circle by double-clicking on it.
- When the circle is selected, you can edit the properties in the toolbox.
- To delete the circle, right-click on it and select **Delete** from the context menu. Alternatively, select the circle and press the **DEL** key.

PROPERTIES

Fill Color - Here you can set the color and opacity for the circle area. The higher the opacity, the lower the transparency, i.e. the less you can see the original chart behind it.

Border Color - Here you can set the border color of the circle.

ARC

The arc is a simple drawing object useful for highlighting special situations in a chart.

You can find the arc tool in the *Chart Tools* group in the toolbar.

ENTERING AN ARC

1. Open a symbol chart.
2. Click on the arc tool in the toolbar.
3. Click into the chart to set the anchor point.
4. Keep the mouse button pressed and draw the arc. Release the mouse button when done.



Arc

EDITING AN ARC

- To move the arc, drag it with the mouse (the mouse pointer turns into a hand symbol) and move it freely. Release the mouse button when done.
- To change the size of the arc, click on an anchor point surrounding the arc and drag it.
- You can duplicate an arc by double-clicking on it.
- When the arc is selected, you can edit the properties in the toolbox.
- To delete the arc, right-click on it and select **Delete** from the context menu. Alternatively, select the arc and press the **DEL** key.

PROPERTIES

Fill Color - Here you can set the color and opacity for the arc area. The higher the opacity, the lower the transparency, i.e. the less you can see the original chart behind it.

Border Color - Here you can set the border color of the arc.

FIBONACCI RETRACEMENTS

Fib. Retracements are a drawing tool with a calculation function. You as the user define the start and end point of the price levels that are to be used for the calculation. You can base your decision on price or correction movements.

Between these two price levels, the retracement lines are drawn according to the numerical proportion of the Fibonacci numerical series. The calculation is based on the golden section. The theory behind using Fib. Retracements for chart analysis is that price movements have rhythms similar to natural processes.



Fib. Retracement

Fib. Retracements may be used for:

- finding course aims of correction movements
- finding course aims of trend movements
- finding price targets of breakouts
- judging the market and its trends
- finding potential entry points at swing points

You can find the tool in the *Chart Tools* group in the toolbar.

ENTERING FIBONACCI RETRACEMENTS

1. Open a symbol chart.
2. Click on the Fib. Retracements tool in the toolbar.
3. Click into the chart to set the start point of the price development (this is the 0% level).
4. Keep the mouse button pressed and move the pointer to the end point of the price development (this is the 100% level). Release the mouse button.

Between the 0 and 100% levels, the calculated lines are drawn. The horizontal position of the anchor points can be changed without effect. Only the distance between the resulting lines is important.

For extrapolating values beyond the 100% level, you can set the corresponding lines to visible (e.g. for 138.2% and 161.8%). To do so, select the check boxes in the property inspector in the toolbar.

DIRECTION-DEPENDENT ANALYSIS

For the sensible usage of the Fib. Retracements, it is important to enter the lines correctly into the chart. The start and end points have to be placed differently depending on whether you wish to analyze correction movements or trend movements.

ANALYSIS OF UPWARDS MOVEMENTS

- For finding possible correction levels in an upwards movement, draw the line from the high to the low point of the movement. This puts the 0% level at the high and the 100% level at the low point. The standard lines that are displayed separate the trend movement into the possible correction levels.
- To find possible future price levels of an upwards movement, however, draw the line from the low to the high point. Then set the levels beyond 100% to visible by selecting them in the properties.



Upwards Movement



Downwards Movement

ANALYSIS OF DOWNWARDS MOVEMENTS

- For finding possible correction levels in a downwards movement, draw the line from the low to the high point of the movement. This puts the 0% level at the low and the 100% level at the high point. The standard lines that are displayed separate the trend movement into the possible correction levels.
- To find possible future price levels of a downwards movement, however, draw the line from the high to the low point. Then set the levels beyond 100% to visible by selecting them in the properties.

EDITING FIB. RETRACEMENTS

- To change the 0% and 100% marks, click the anchor point of the respective line (the mouse pointer changes into an x-y axis symbol) and move it. The line positions will be adapted accordingly.
- To move Fib. Retracements, drag them with the mouse (the mouse pointer turns into a hand symbol) and move it freely. Release the mouse button when done.
- You can edit the ratio of each Fib. line in the retracements. To do so, click on the numbers next to the check boxes (in the properties) and enter new values.
- You can duplicate the Fib. Retracements by double-clicking on them.
- When the Fib. Retracements are selected, you can edit the properties in the toolbox.
- The alert function is available for Fib. Retracements. To set an alert, set the **Alert Mode** from *Disabled* to one of the options (see below).
- To delete the Fib. Retracements, right-click on them and select **Delete** from the context menu. Alternatively, select the Fib. Retracements and press the **DEL** key.

PROPERTIES

FIB. RETRACEMENTS PROPERTIES

% 1, % 2... - Select from the list of available Fib. levels the ones you want to display. You can edit the values by clicking on the numbers to the right of the check boxes and entering new ones.

Extend Left/Extend Right - Select whether the lines are to be extended to the left and/or right until the edge of the chart.

Modus - Select whether the lines are to be displayed in linear or logarithmic scale.

GRAPHIC PROPERTIES

The following properties can be set for the Fib. Retracements tool:

Color - Here you can select a line color.

Width - Here you can set a width for all lines by entering a number.

Style - Here you can select a line style, e.g. solid or dotted.

Labels - Here you can set the value label for the lines to visible or hidden. In Tradesignal you can also specify the location at which the label should be displayed.

Show Guide Line - Here you can select whether the guide line (the line connecting the two anchor points of the 0% and 100% lines) is visible or hidden.

ALERT PROPERTIES

The Retracements tool can be used with various kinds of alerts.

Alert Direction - Here you can select from the list whether crossings of all types lead to alerts or only crossings in a certain direction.

Alert Mode - Here you can select an alert mode from the list. Select whether crossing raises an alert always, once per bar or once overall.

Alert Name - Here you can specify a name for alerts generated by this tool. It will be included in the signal field of the alert.

Alert Trigger - Here you can select when an alert should be triggered.

- *Crossed*: Triggers when the price crosses the tool, e.g. goes from below the tool to above the tool

- *Hit or Crossed*: Same as above, but also triggers when the price merely hits the tool. This setting is useful for a Point&Figure Chart.

For further information on alert modes, see Alerts.

FIBONACCI EXTENSIONS

Fibonacci Extensions is a drawing tool with a calculation function. You as the user define the start on a significant swing low, then drag your mouse and click on the most recent swing high. Finally, drag your mouse down and click on any of the retracement levels.

Between these three price levels, the extension lines are projected according to the numerical proportion of the Fibonacci numerical series. The calculation is based on the golden section. The theory behind using Fibonacci Extensions for chart analysis is that price movements have rhythms similar to natural processes.



Fibonacci Extensions

Most traders use Fibonacci Extensions in combination with other technical indicators/patterns to help them determine appropriate target prices.

You can find the tool in the *Chart Tools* group in the toolbar.

ENTERING FIBONACCI EXTENSIONS

1. Open a symbol chart.
2. Click on the Fibonacci Extensions tool in the toolbar.
3. Click into the chart to set the low swing point.
4. Keep the mouse button pressed and move the pointer to a swing high. Release the mouse button.

The third point will be automatically calculated, but may be repositioned later by clicking and dragging it. For extrapolating values beyond the 100% level, you can set the corresponding lines to visible (e.g. for 138.2% and 161.8%). To do so, select the check boxes in the property inspector in the toolbar.

DIRECTION-DEPENDENT ANALYSIS

For the sensible usage of the Fibonacci Extensions, it is important to enter the lines correctly into the chart. The start and end points have to be placed appropriately depending on whether you wish to analyze correction movements or trend movements.

EDITING FIBONACCI EXTENSIONS

- To change the marks, click the anchor point of the respective line (the mouse pointer changes into an x-y axis symbol) and move it. The line positions will be adapted accordingly.
- To move Fibonacci Extensions, drag them with the mouse (the mouse pointer turns into a hand symbol) and move it freely. Release the mouse button when done.

- You can edit the ratio of each Fibonacci line in the extension. To do so, click on the numbers next to the check boxes (in the properties) and enter new values.
- You can duplicate the Fibonacci Extensions by double-clicking on them.
- When the Fibonacci Extensions is selected, you can edit the properties in the toolbox.
- The alert function is available for Fibonacci Extensions. To set an alert, set the **Alert Mode** from *Disabled* to one of the options (see below).
- To delete the Fibonacci Extensions, right-click on them and select **Delete** from the context menu. Alternatively, select the Fibonacci Extensions and press the **DEL** key.
- You can extend the Fibonacci lines by either clicking and dragging an end on either side, or enabling the *Extend Left* or *Extend Right* properties in the Inspector.

PROPERTIES

FIBONACCI EXTENSIONS PROPERTIES

% 1, % 2... - Select from the list of available Fib. levels the ones you want to display. You can edit the values by clicking on the numbers to the right of the check boxes and entering new ones.

Extend Left/Extend Right - Select whether the lines are to be extended to the left and/or right until the edge of the chart.

Mode - Select whether the lines are to be displayed in linear or logarithmic scale.

GRAPHIC PROPERTIES

The following properties can be set for the Fibonacci Extensions tool:

Color - Here you can select a line color.

Width - Here you can set a width for all lines by entering a number.

Style - Here you can select a line style, e.g. solid or dotted.

Labels - Here you can set the value label for the lines to visible or hidden. In Tradesignal you can also specify the location at which the label should be displayed.

Show Guide Line - Here you can select whether the guide line is visible or hidden.

ALERT PROPERTIES

The Extensions tool can be used with various kinds of alerts.

Alert Direction - Here you can select from the list whether crossings of all types lead to alerts or only crossings in a certain direction.

Alert Mode - Here you can select an alert mode from the list. Select whether crossing raises an alert always, once per bar or once overall.

Alert Name - Here you can specify a name for alerts generated by this tool. It will be included in the signal field of the alert.

Alert Trigger - Here you can select when an alert should be triggered.

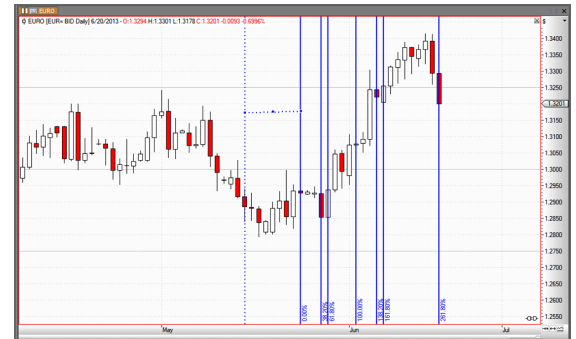
- *Crossed*: Triggers when the price crosses the tool, e.g. goes from below the tool to above the tool
- *Hit or Crossed*: Same as above, but also triggers when the price merely hits the tool. This setting is useful for a Point&Figure Chart.

For further information on alert modes, see Alerts.

FIBONACCI TIME PROJECTION

Fib. Time Projection is a drawing tool with a calculation function. You as the user define two anchor points at extremes, i.e. either high or low points in the chart.

The Fib. Time Projection then extends the distance between the two anchor points in the time line with a number of vertical lines. The lines are calculated by multiplying the distance between the anchor point with the Fibonacci numbers. The result is a number of projection lines that can be useful for finding swing points in charts.



Fibonacci Time Projection

Fib. Time Projections may be used for:

- finding possible swing points
- finding time aims for current price movements

You can find the tool in the *Chart Tools* group in the toolbar.

ENTERING FIBONACCI TIME PROJECTION

1. Open a symbol chart.
2. Click on the Fib. Time Projection tool in the toolbar.
3. Click into the chart on a high or low point to set the first anchor point.
4. Keep the mouse button pressed and move the pointer to another high or low point of a similar price level in the chart. Release the mouse button.

In a time projection, the Fib. lines are drawn to the right of the anchor points. The vertical position of the anchor points can be changed without effect. Only the distance between the anchor points is important.

For extrapolating values beyond the 100% level, you can set the corresponding lines to visible (e.g. for 138.2% and 161.8%). To do so, select the check boxes in the property inspector in the toolbar.

EDITING FIB. TIME PROJECTION

- To move the anchor points, click on the respective point (the mouse pointer changes into an x-y axis symbol) and move it. The line positions will be adapted accordingly.
- To move the Fib. Time Projection, drag it with the mouse (the mouse pointer turns into a hand symbol) and move it freely. Release the mouse button when done.
- You can edit the ratio of each Fib. line in the retracements. To do so, click on the numbers next to the check boxes (in the properties) and enter new values.
- You can duplicate the Fib. Time Projection by double-clicking on it.

- When the Fib. Time Projection is selected, you can edit the properties in the toolbox.
- To delete the Fib. Time Projection, right-click on it and select **Delete** from the context menu. Alternatively, select the Fib. Time Projection and press the **DEL** key.

PROPERTIES

FIB. TIME PROJECTION PROPERTIES

% 1, % 2... - Select from the list of available Fib. levels the ones you want to display. You can edit the values by clicking on the numbers to the right of the check boxes and entering new ones.

GRAPHIC PROPERTIES

The following properties can be set for the Fib. Time Projection tool:

Color - Here you can select a line color.

Width - Here you can set a width for all lines by entering a number.

Style - Here you can select a line style, e.g. solid or dotted.

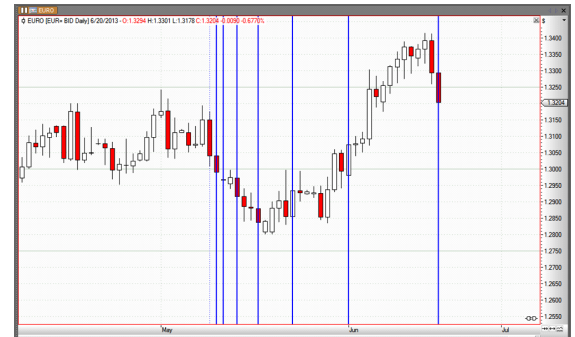
Label Placement - Here you can set whether and where the value labels for the lines will be displayed, bottom or top.

Show Guide Line - Here you can select whether the guide line (the line connecting the two anchor points) is to be visible or hidden.

FIBONACCI TIME ZONES

Fib. Timezones is a drawing tool with a calculation function. You as the user define two anchor points at extremes, i.e. either high or low points in the chart.

The Fib. Timezones then extends the distance between the two anchor points in the time line with a number of vertical lines. The lines are calculated by multiplying the distance between the anchor point with the Fibonacci numbers. The result is a number of timezone lines that can be useful for finding swing points in charts.



Fibonacci Timezones

Fib. Timezones may be used for:

- finding possible swing points
- finding time aims for current price movements

You can find the tool in the *Chart Tools* group in the toolbar.

ENTERING FIBONACCI TIMEZONES

1. Open a symbol chart.
2. Click on the Fib. Timezones tool in the toolbar.
3. Click into the chart on a high or low point to set the first anchor point.
4. Keep the mouse button pressed and move the pointer to another high or low point of a similar price level in the chart. Release the mouse button.

In a timezone projection, the Fib. lines are drawn to the right of the anchor points. The vertical position of the anchor points can be changed without effect. Only the distance between the anchor points is important.

For extrapolating values beyond the 100% level, you can set the corresponding lines to visible (e.g. for 138.2 and 161.8%). To do so, select the check boxes in the property inspector in the toolbar.

EDITING FIB. TIMEZONES

- To move the anchor points, click on the respective point (the mouse pointer changes into an x-y axis symbol) and move it. The line positions will be adapted accordingly.
- To move the Fib. Timezones, drag it with the mouse (the mouse pointer turns into a hand symbol) and move it freely. Release the mouse button when done.
- You can edit the ratio of each Fib. line in the retracements. To do so, click on the numbers next to the check boxes (in the properties) and enter new values.
- You can duplicate the Fib. Timezones by double-clicking on it.

- When the Fib. Timezones is selected, you can edit the properties in the toolbox.
- To delete the Fib. Timezones, right-click on it and select **Delete** from the context menu. Alternatively, select the Fib. Timezones and press the **DEL** key.

PROPERTIES

The following properties can be set for the Fib. Timezones tool:

Color - Here you can select a line color.

Width - Here you can set a width for all lines by entering a number.

Style - Here you can select a line style, e.g. solid or dotted.

FIBONACCI ARCS

Fib. Arcs are a drawing tool with a calculation function. You as the user define the start and end point of the price levels that are to be used for the calculation. The Fib. Arcs are then calculated and drawn, similar to the Fib. Retracements. The theory behind using Fibonacci numbers for chart analysis is that price movements have rhythms similar to natural processes.

You may use Fibonacci Arcs for

- finding consolidation aims in price movements
- finding resistance and support levels

You can find the tool in the *Chart Tools* group in the toolbar.



Fibonacci Arcs

ENTERING FIBONACCI ARCS

1. Open a symbol chart.
2. Click on the Fib. Arcs tool in the toolbar.
3. Click into the chart, for example a high point.
4. Keep the mouse button pressed and move the pointer to the next low point. Release the mouse button.

Between the anchor points, the calculated arcs are drawn.

For extrapolating values beyond the 100% level, you can set the corresponding arcs to visible (e.g. for 138.2 and 161.8%). To do so, select the check boxes in the property inspector in the toolbar.

EDITING FIB. ARCS

- To move the baseline, click the respective anchor point (the mouse pointer changes into an x-y axis symbol) and move it. The arcs will be adapted accordingly. Depending on the position of the middle to the outer anchor point, the arcs are drawn upwards or downwards.
- To move Fib. Arcs, drag them with the mouse (the mouse pointer turns into a hand symbol) and move them freely. Release the mouse button when done.
- You can edit the ratio of each Fib. line in the arcs. To do so, click on the numbers next to the check boxes (in the properties) and enter new values.
- You can duplicate the Fib. Arcs by double-clicking on them.
- When the Fib. Arcs are selected, you can edit the properties in the toolbox.
- To delete the Fib. Arcs, right-click on them and select **Delete** from the context menu. Alternatively, select the Fib. Arcs and press the **DEL** key.

PROPERTIES

FIB. ARCS PROPERTIES

% 1, % 2... - Select from the list of available Fib. levels the ones you want to display. You can edit the values by clicking on the numbers to the right of the check boxes and entering new ones.

GRAPHIC PROPERTIES

The following properties can be set for the Fib. Arcs tool:

Color - Here you can select a line color.

Width - Here you can set a width for all lines by entering a number.

Style - Here you can select a line style, e.g. solid or dotted.

FIBONACCI FANS

Fib. Fans are a drawing tool with a calculation function. You as the user define the start and end point of the price levels that are to be used for the calculation. The Fib. Fans are then calculated and drawn, similar to the Fib. Retracements. The theory behind using Fibonacci numbers for chart analysis is that price movements have rhythms similar to natural processes.

You may use Fib. Fans for

- finding support lines in upwards movements
- finding resistance levels in downwards movements

You can find the tool in the *Chart Tools* group in the toolbar.



Fibonacci Fans

ENTERING FIBONACCI FANS

1. Open a symbol chart.
2. Click on the Fib. Fans tool in the toolbar.
3. Click into the chart to set an extreme point, for example at a high.
4. Keep the mouse button pressed and move the pointer to the next diametrical extreme, for example a low point. Release the mouse button.

Between the anchor points, the calculated fans are drawn. They can be seen as resistance and support levels.

For extrapolating values beyond the 100% level, you can set the corresponding fans to visible (e.g. for 138.2% and 161.8%). To do so, select the check boxes in the property inspector in the toolbar.

EDITING FIB. FANS

- To move the baseline, click the respective anchor point (the mouse pointer changes into an x-y axis symbol) and move it. The fans will be adapted accordingly. Depending on the position of the middle to the outer anchor point, the fans are drawn upwards or downwards.
- To move Fib. Fans, drag them with the mouse (the mouse pointer turns into a hand symbol) and move them freely. Release the mouse button when done.
- You can edit the ratio of each Fib. line in the fans. To do so, click on the numbers next to the check boxes (in the properties) and enter new values.
- You can duplicate the Fib. Fans by double-clicking on them.
- When the Fib. Fans are selected, you can edit the properties in the toolbox.
- To delete the Fib. Fans, right-click on them and select **Delete** from the context menu. Alternatively, select the Fib. Fans and press the **DEL** key.

PROPERTIES

FIB. FANS PROPERTIES

% 1, % 2... - Select from the list of available Fib. levels the ones you want to display. You can edit the values by clicking on the numbers to the right of the check boxes and entering new ones.

GRAPHIC PROPERTIES

The following properties can be set for the Fib. Fans tool:

Color - Here you can select a line color.

Width - Here you can set a width for all lines by entering a number.

Style - Here you can select a line style, e.g. solid or dotted.

CHART

Charts are graphical displays of prices for stocks, commodities, indices or funds. They can be used for:

- Supporting trading decisions by following price development.
- Making trading decisions solely on the basis of price development.
- Gathering information about possible future price developments.
- Preparing analysis for past price developments.
- Visually following symbol positions.
- Systematically developing signals for portfolio trading.
- Graphically displaying trading system results.
- Graphically displaying mathematical and statistical calculations.
- Visualizing the performance development of a symbol.



A Chart

With the help of simple methods it is possible to process price developments from the past to allow prognosis for the future. These methods are known by the term "chart analysis".



Charts have the following design parameters:














- The maximum number of tools per charts is 250 (e.g. text elements, chart tools).
- The maximum history length is 500000.
- The maximum number of instruments per chart is 12 (including the subcharts, e.g. 6 instruments with one indicator each).

For additional information, also see these related topics:

- In the chapters Using Indicators and Using Strategies, you can find information on how to add these to the chart for chart analysis. It is also possible to change their output using the available chart types.
- In the chapter Chart Tools you find information on how to add and use these drawing tools for your chart analysis.
- In the chapter Timespans and Periods you can find information on how to toggle quickly between different trading periods, how to define your own periods and how to set the visible time span and which pre-defined time spans are available.
- In the chapter Chart Properties you can find information about layout settings that define the appearance of the chart and the displayed symbols. You can also find out about styles and templates that allow you saving your own configuration and applying them to other charts.
- In the chapter Publishing you can find information on how to print your chart, save an image of it, or send it via E-mail or Tradesignal Online forum to other users.

AVAILABLE CHART TYPES

Symbol	Name	Description
	Bar Chart	Plots price movement; vertical bars indicate range; markers on the bar show open and close prices
	Candlestick Chart	Price chart that shows the open, high, low, and close prices over a given time period

	Line Chart	Plots a line connecting consecutive prices
	Stepped Line Chart	Similar to a line chart where price levels are connected to form a sequence of discrete steps
	Symbol Chart	Plots symbols at each price
	Filled Area Chart	Plots a filled region between two prices
	Forest Chart	Plots a histogram between prices by time
	Linked Forest Chart	A forest chart with a line joining the columns
	Candle Volume Chart	A candlestick chart with the width of each candle representing the volume
	Equi Volume Chart	Plots vertical bars showing price movement with the widths representing the volume
	Kagi Chart	Plots lines with different line width, independent of the time line
	Renko Chart	Plots a sequence of ascending and descending bricks to indicate price movement, independent of the time line
	Three Line Break Chart	Plots a sequence of bricks depending on close prices, independent of the time line
	Heikin Ashi Chart	Similar to candlesticks, but uses averaged prices
	Point&Figure Chart	Plots price increases and decreases as a series of stacked Xs and Os, independent of the time line


Market Profile

Inserts a market profile chart (various price and turnover data) in the workspace

WAYS OF CALLING UP A CHART

There are three main routes for calling up a chart for a commercial paper:

- If you know the ticker symbol, enter it in the command line. (Tip: You can find ticker symbols in the Symbol Search.)
- If you don't know the ticker symbol of the commercial paper, you can find it in the symbol lists.
- If you receive an E-Mail with charts, you can save them and call up the charts.

You can also open symbols in subcharts.

CALLING UP A CHART VIA THE COMMAND LINE

1. Type the ticker symbol (e.g. CAT) in the command line.
2. Press ENTER. Alternatively, you can choose the entry **New Chart** from the pull-down menu.

CALLING UP A CHART FROM A SYMBOL LIST

For this you need to have a symbol list, e.g. one for the USA Dow Jones.

1. Click on the yellow star symbol in the lower area of the tool bar. The symbol list manager opens.
2. Double-click on the symbol list to open it. There are two ways to open a chart for a list entry:
 1. If the working area doesn't display a chart, or if you want to substitute the symbols in an existing chart, double-click on the list entry.
 2. If you want to open a new chart, right-click on the list entry and choose **New Chart** from the context menu.

Now you can use the two green arrow buttons right next to the command line to move through the list and get each entry displayed in the chart.

Tip: You can also open the symbol list as a table in the Scanner, the Watchlist and as Portfolio and call up a new chart by right-clicking on a table entry and choosing **New Chart** from the context menu.

CALLING UP A CHART FROM AN E-MAIL

Tradesignal offers you the option to send charts or whole workspaces by E-Mail. In a similar way, you can import these sent objects into the program.

1. If you receive an E-mail with a workspace, save it locally. We recommend that you choose the folder *Tradesignal Files* which is located in *Personal Folder*.
2. Update the list of workspaces by right-clicking in the toolbar on the folder *Tradesignal Files* and choosing **Update**. The saved workspace appears in the list of workspaces and can be opened from there.

COPY, PASTE AND DELETE CHARTS

You can edit charts with the functions in the *Edit* group in the toolbar:

- **Copy** - Copy the current chart configuration into your Windows clipboard.
- **Paste** - Paste the copied chart configuration into a currently opened chart, a new workspace, an office document or an image processing application.
- **Delete** - Delete the currently active chart.

NAVIGATION IN THE CHART

To activate a chart, click on its tab. In the active chart, the small plug symbol on the lower right of the chart or a portfolio is visible, see the chapter Automatic Order Routing.

The axes and visible areas can be set up relatively flexible. To return to the default settings, you can either press the key **F8** or click on the axes symbol on the lower right of the chart. More information you can find in the chapter Timespans and Periods.

MOVE (SCROLL) AXES

You can move (scroll) the x- and y-axes (usually time and price axes) by clicking on them with the mouse and keeping the mouse button pressed while moving it up/down or left/right.

Alternatively, you can scroll the axes by pressing the cursor keys (up/down, left/right). To speed up the scrolling, press the **STRG** key in addition to the cursor keys.

STRETCH/SHRINK AXES

You can change the scale of the axes, so that more or less of the range is shown. To do this, click on it with the right mouse button and keep it pressed while moving the mouse left/right or up/down.

Alternatively, keep the **SHIFT** key pressed while pressing the cursor keys (up/down, left/right).

USE THE X-AXIS SCROLLBAR

You can also scroll the x-axis by using the scrollbar, which is situated on the lower right corner, to the left of the axes symbol. Click on it and keep the mouse button pressed to drag the scrollbar to the left or right. The scroll speed depends on how much the time axis is currently stretched or shrunk.

PANORAMIC SCROLLBAR



Chart with shortened axes

In Tradesignal you can use the panoramic scrollbar to move (scroll) the x-axis by dragging it or clicking on the axis itself. The background always shows all available data as a filled area chart to give you additional information about your current viewport. It is possible to stretch/shrink (zoom) the axis by dragging the grippers of the new panoramic scrollbar. New buttons replacing the axes button makes it easy to toggle between normal and panoramic scrollbar or show recent/all data. In the chart properties you can change the height of the panoramic scrollbar or switch to the normal one.



Chart with panoramic scrollbar

CHANGE SCALE ALIGN

You can change the scale align for the y-axis via the properties inspector (left, right, left/right), or via the chart legend (left and right).

For the latter, click on the chart legend and keep the mouse button pressed while dragging it to the left or right side of the chart, until a little arrow symbol appears. Then release the mouse button. This way, you can drag and drop the scale to the left or right side of the chart.

USE THE ZOOM FUNCTION

If you want to take a closer look at certain chart regions, price patterns or signals of trading systems, the Zoom function is the right tool.

1. Activate the Zoom tool by clicking on the small icon with a magnifying glass and a plus symbol in the *Chart* area of the toolbar.
2. Click on the first corner of the area you want to enlarge.
3. Draw the rectangle around the area.
4. Release the mouse button. The area is zoomed.

To reset the Zoom and return to the previous view, click into an empty area of the chart with the activated Zoom tool.



Chart with zoomed area

To deactivate the Zoom tool, click on another tool like the select tool or target cursor. Alternatively, right-click into the chart and then click with the left mouse button into the empty area next to the context menu. This way, you return to the select mode. Although the second variant may sound complicated, it's a lot faster than the first one.

USE THE TARGET CURSOR

To retrieve detailed information about the region below the mouse cursor, you can use the Target Cursor.

Activate the Target Cursor by clicking on the small cross-hairs icon in the *Chart* area of the toolbar. When you click into the chart now, information about that position, the trading period, prices, indicators, etc. is displayed in a pop-up window. This window remains visible as long as the Target Cursor is active.



Using the Target Cursor

To return to the select mode, click on the select mode button in the toolbar.

Alternatively, right-click into the chart and then click with the left mouse button into the empty area next to the context menu. This way, you return to the select mode. Although the second variant may sound complicated, it is a lot faster than the first one.

You can change the appearance of the Target Cursor in the chart properties, **Target Cursor Style**. Select **Thin Line** or **Whole Column**.

To have a temporary "target cursor" with an information pop-up, click and keep pressed the left mouse button in the select mode. The mouse pointer changes into a target cursor and a pop-up window with information appears. The window disappears when you release the mouse button.

CONTEXT MENUS IN THE CHART

If you right-click on an element in the chart, a context menu opens, offering you various functions for this element. The most important functions are:

- Delete the selected element
- Show properties of the selected element
- For the symbol: starting the price editor and trading time manager, changing the period, saving chart as template, report bad ticks, trade symbol (automatically opens the order bar).
- For the indicators / trading systems: starting the program editor, request online information.

SUBCHARTS

Subcharts are a special form of chart. They can be recognized by not having their own tab. Subcharts can have the same date axis or a different one than the main chart. As subcharts, the following documents can be displayed: symbols, indicators, strategies, equity curves (e.g. of the portfolio).

OPENING A SUBCHART

To open an instrument as a subchart, proceed as follows:

1. Click on the symbol button in the toolbox to open the symbol lists manager.
2. Double-click on the symbol list to open it.

3. Right-click on a symbol in the list to open the context menu. You have two options:

- - If you want to open the subchart with the same time axis as the main chart, select **Insert Symbol** from the context menu.
 - If you want to open the subchart with its own time axis, select **Insert Symbol with new Date Scale** from the context menu. (The y-axes are always independent of each other.)

The subchart opens below the main chart in the same tab.

Note that this action also depends on the option **Open symbols in their own subchart**. This defines how the program reacts to **Insert Symbol**. If the option is selected, symbols are opened in a subchart. Otherwise, the symbol gets added to the existing chart.

CLOSING A SUBCHART

To close a single subchart, click on the **x** button in the upper right corner of the subchart. Alternatively, right-click into the subchart and select **Delete subchart** from the context menu.

When you close a chart, all its subcharts are closed automatically.

NAVIGATION IN SUBCHARTS

- To maximize a subchart, click on the button with the window symbol in the upper right corner of the subchart. Two little arrow buttons appear (right/left). Click on them to move between subcharts.
- To display all subcharts again, click on the button with the staggered windows.

To activate a subchart, click on the subchart legend. In the active subchart, the small plug symbol is visible on the lower right, just like for the chart. (See the chapter Automatic Order Routing).

EDITING SUBCHART PROPERTIES

When you change the trading period for a chart via the **Period** button of the command line (see the chapter Timespans and Periods), the period for all subcharts is changed accordingly.

To change the period for a single subchart, right-click into the subchart and select another **Period for Object** in the context menu.

MOVING INSTRUMENTS BETWEEN (SUB-)CHARTS

If an instrument is opened in a chart and you would like to move it into a subchart, you can drag and drop the instrument.

1. Click on the instrument in the legend and drag the instrument down. A little down-pointing arrow is added to the mouse cursor.
2. Drop the legend into another subchart or pull it down until the edge of the window, so that a new subchart is opened.

In the same way, you can move instruments from one subchart into another or into the main chart. This way, you can display two symbols together in one chart even though the option **Open Symbols in their own Subchart** in the Tradesignal Options is active.

INDICATORS AND STRATEGIES IN SUBCHARTS

Whether an indicator is opened in a subchart depends on their Equilla programming. You can move indicators by dragging them into other (sub-)charts, as described above. To which instrument an indicator is applied to is given in the indicator legend and is independent from the arrangement on the user interface. (See also the chapter Using Indicators.)

For strategies, the same rules apply in principle. However, usually their graphic output are only a few arrows at the instrument curve. As this doesn't make not much sense in a subchart, most strategies are opened in the same (sub-)chart as the instrument they are applied to.

SET NEW DEFAULT CHART STYLE

When Tradesignal is first installed, the default chart style is the candlestick chart. However, you can also set the bar and line chart as default. Proceed as follows:

1. Open the File menu.
2. Select the **Tradesignal Options**.
3. Open the **Chart** settings.
4. In the *Default Chart Style* area, select whether you want the candlestick, line, or bar chart as standard. Confirm the new settings by clicking **OK**.

As of now, all new charts are displayed according to the new default setting.

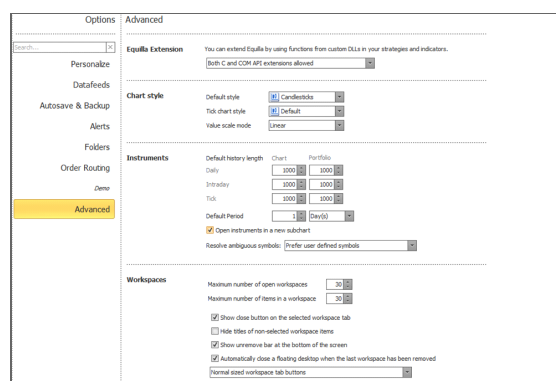


Chart settings in the Tradesignal Options

COMPARING SYMBOLS

If you want to compare several symbols in the chart, it may be essential to normalize their measuring unit to get meaningful results. Tradesignal offers you different possibilities to achieve this goal. The following examples use Tradesignal Realtime as data provider:

CURRENCY CONVERSION

If you are trading the same symbol on different exchanges you may encounter the problem that they are displayed in different currencies. The following example shows you how to unify the currency in the chart:

1. Open a chart with Microsoft, **MSFT NAS**, a symbol traded on the American market in *US Dollars*.
2. Add another Microsoft symbol traded on a different exchange, e.g. **MSF GER**, which is traded in Germany in *Euro*.
3. Select the symbol traded in Germany in the chart and change the currency to **US Dollars** in the instrument properties.
4. Select the first Microsoft symbol and add the **Spread Diff** indicator from the Tradesignal package. In the dialog, choose the symbol traded in Germany as second input.

Please note that the currency conversion is not available for all symbols, such as indices which are displayed in index points. The

list of available currencies depends on the selected data provider.

UNIT CONVERSION

Unit conversion can be interesting for comparing symbols that are traded in different measurement units, e.g. barrel, bushel, tons, etc. This information and the list of available units and used abbreviations depends on the data provider .

Example for soy beans and soy oil:

1. Create a Continuous Contract for soy beans and choose **S** as Future Root Code. The future is traded in US Cent per **Bushel (Bsh)**.
2. Create another Continuous Contract for soy oil and choose **BO** as Future Root Code. This one is traded in US Cent per **Pound (Lbs)**.
3. Add both UDCs to a chart and select the soy oil symbol.
4. Change the unit in the instrument properties from **Lbs** to **Bsh**.
5. Add one of the **Spread** indicators from the Tradesignal package to display the difference.

DISPLAY UNIT

Symbols can have different display units which refer to the displayed numbers of decimal places. For example, the Dow Jones index has two decimal places while the Euro has four decimal places. This is how you normalize the display of symbols in the chart:

1. Create a chart with the Dow Jones, **.DJI**.
2. Add the Euro (**EURUSD**) symbol to the chart and select it.
3. Open the properties and change the instrument property **Display Unit** to **0.01** to display the Euro with two decimal places.

The default value is provided by the data provider. It can be changed in the session properties for each symbol by opening the Session Manager via the context menu or the toolbar.

Tip: You can change the display unit also for indicator and strategy outputs. Select the element in the chart and change the value in the properties. The default value of the output is based on the underlying symbol display unit.

The display unit is also available in the Scanner, Watchlist and Portfolio and can be used to align the decimal point for all symbols below each other in the chart.

SEASONAL CHARTS

With the seasonal mode in the chart properties, Tradesignal offers you the ability to overlay the prices of different years for one or more symbols. Comparing the symbols this way emphasizes seasonal pattern, e.g. if used with commodities that have seasonal variations. See the article Seasonal Chart for more details

PERCENT PERFORMANCE CHARTS

You can display the price movements of a symbol as percentage increase/decrease relative to a start price by using percent performance transformation. This makes it easier to compare the performance of symbols since their absolute price values are

converted to relative price movements. See the article Percent Performance for more details.

SEMI-LOGARITHMIC DISPLAY

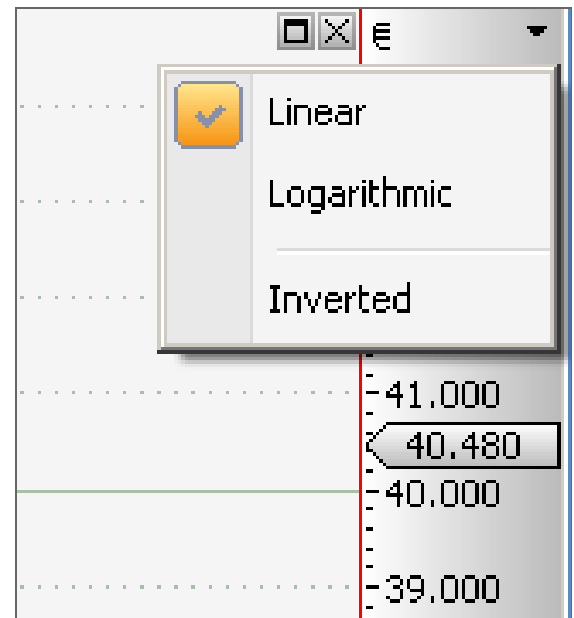
The value axis can be displayed using a logarithmic scale. This is mainly used for symbols with a long history, to emphasize fluctuations in prices. The changes are not displayed linearly but perceptually, therefore the increase from 1 to 10 is identical to the increase from 10 to 100.

Example for changing the Dow Jones display from linear to logarithmic:

1. Open a chart with the Dow Jones, '.DJI.'
2. Open the context menu for the symbol (see picture) and choose **Logarithmic**. To return to the linear display, repeat this action but select **Linear** instead.

You also have the option of inverting the value scale by selecting **Inverted**, this will cause the lowest values to be displayed at the top.

Please note that the left value axis can be separately changed in display if you have a second symbol in the chart.



Context menu of value scale

CHART PROPERTIES

Just as every chart analysis is governed by the individual preferences of each trader, the chart properties like colors, fonts and other optical settings can be adjusted to personal likings. You can use the chart properties and the styles and templates functions to do the following:

- Adapt the charts to personal color and font preferences
- Optimize charts for usage in print or web media
- Re-use the settings for other charts
- Enhance the recognition factor of your work in publications
- Define and use templates for the chart configuration
- Define and use Style properties for Charts and other documents like Scanners, Watchlists or Portfolios.



A Chart

SETTING THE CHART PROPERTIES

As "Chart Properties", the following options should be included here:

- You can edit the chart properties. To do this, open the toolbox and click on **Chart** in the upper region of the property inspector to see the parameters for axes, grid, legend etc. In case of strategies, more settings are available.
- Some settings of the symbols are also reflected in the chart. To access these settings, click on the symbol in the upper region of the property inspector. The parameters for volume and history length are available.
- You can set up the optical chart properties by using styles (pre-set or your own). If you have saved templates, you can also apply those to new charts.

DISPLAYING THE PROPERTIES OF SELECTED ITEMS

When you select an item in Tradesignal, its properties are displayed in the toolbox. For example, when you click on a chart, the chart properties are displayed; when you click on a symbol, the symbol properties are displayed.

This is the default setting. However, you can change this by changing the setting for **Automatically show properties when an item is selected** in the *Advanced* area of the Tradesignal Options.

SAVING, RESTORING AND RESETTNG DEFAULT SETTINGS

On top of the chart properties in the properties manager, to the right, you can find two important buttons.

- Disk button (Save as Default) - Click to save your current settings as the new default settings.
- Restore button (Restore Default Settings, circled arrow) - Click to reset all settings to the default settings.

Not all settings can be saved like this:

- Some chart settings are done in the Tradesignal Options, like the history length, the default chart type etc.
- The standard period and other user interface settings are entered in the Tradesignal Options.

Some settings also do not make much sense as a default, like the value for a stop line.

To restore the original default settings of your Tradesignal installation, click on the **Restore Default Settings** button in the Tradesignal Options.

CHART PROPERTIES IN THE TOOLBOX

Beside the color selectors for the background, date and value axis, you can also find parameters for the grid and the chart legend.

CHART

Font - Here you can select a font type (Tahoma, Arial...), font style (bold, italic...) and font size. Click on the entry to open the dialog in which you can edit the values.

Auto-Hide Scales - If activated, the scales are hidden if the chart is displayed in a very small size.

Background - Open a dialog in which you can set a background color. You can choose between solid colors or gradients (vertical or horizontal). To create a gradient, you need to select two colors. For each of them, a color palette is available.

Show Gaps - Select whether times without trading should be invisible or visible in the chart.

History Length - Enter the length of the price history here. The maximum length in Tradesignal is 500000 data points, which effectively means the length is limited by your data provider.

Refresh on Historic Updates - Set to **Yes** if the indicators/strategies should be recalculated when changing the data history.

Min. Referenced Bars - Enter the minimum number of bars that should be used for calculating the values of indicators and strategies.

Target Cursor Style - Here you can edit the style of the target cursor, **Thin Line** or **Whole Column**.

Replace Symbols - Control how a symbol replace operation is performed with regard to unit and currency conversion.

Template - Shows if this chart is currently linked to a template file.



Chart in the property inspector

DATE AXIS

For the date axis, you can set the background color and font properties. In addition, you can also set whether the scrollbar should be active. In Tradesignal you can toggle between the panoramic scrollbar and the normal one or adjust the height of the panoramic scrollbar.

VALUE AXIS

For the value axis, you can set the background color and font properties. In addition, you can set the axis to invisible to have more room for your graph.

GRID

A grid can be set up in the background of the chart. You can define the grid properties like line art, style (points, vertical, horizontal...), and mode of the grid (plain or every second column colored). You can also set the background and column colors.

LEGEND

Here you can define the font settings and the way symbols, indicators and strategies are to be displayed in the legend. The available parameters depend on the selected chart type. You also have the option to display net and percent change (relative to previous bar).

PERCENT PERFORMANCE

This allows you to set global percent performance properties which are used by all symbols in your chart. Percent performance charts allow you to easily compare the relative performance of multiple symbols.

RISK ORDER APPEARANCE

If strategies are applied to the graph, you can define the optical qualities of the trading system signals here. You can set the colors of the arrows (pointing at events in the curve) and also define how much information should be displayed for the trades.

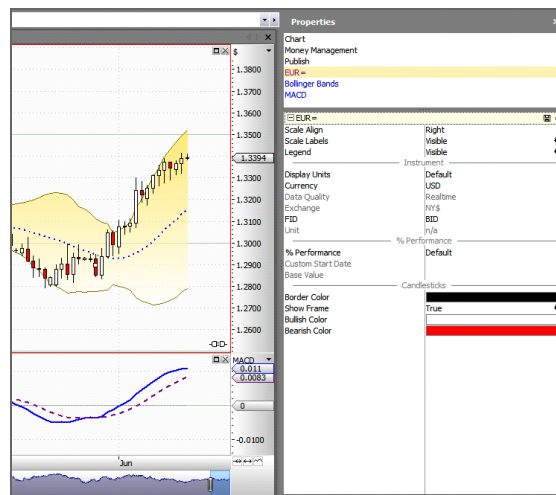
SYMBOL PROPERTIES IN THE TOOLBOX

Click on the symbol name on top of the property inspector in the toolbox to edit the symbol properties. Depending on the symbol in the chart, you can do the following:

- Display the trading volume and define colors for rising/falling
- Define the colors of candles, bars etc. in the chart
- Set other properties, e.g. "Open Interests" for Futures

STANDARD PROPERTIES OF STOCKS IN CHARTS

In the property inspector of the displayed stock you can find the following parameters.



Symbols in the property inspector

Scale Properties

Scale Align - Select whether scales should be displayed left, right, or left and right.

Scale Labels - Select whether the current price should be displayed as a special label on the scale.

Legend - Here you can set the legend (top left of the chart) to either visible or hidden (invisible).

Volume Properties

Volume Bars - Select whether the volume should be displayed as a forest (histogram) chart on the bottom of the chart.

Rising / Falling Color - Here you can set the colors for the volume bars.

Instrument Properties

Depending on the chart type, additional parameters may be available.

Display Units - Here you can change the display units, e.g. with how many decimal places or fractions the prices should be displayed. The default display unit will be read from the symbol properties. You can customize the default display unit via sessions.

Currency - By default, the currency related to the exchange on which the symbol is traded will be displayed; you can change the default currency to your preferred one.

Data Quality - Displays whether the data is realtime or delayed.

Exchange - Displays the exchange at which the symbol is traded.

FID - Here you can set which fields (Last, Bid, Ask) are used for chart calculation.

Unit - Here you can change the unit, e.g. Barrel, Bushel etc. Whether this option is available depends on the used data provider.

PERCENT PERFORMANCE

In this group are percent performance properties for a specific symbol. This allows you to enable it only for certain symbols.

*[CHART TYPE]*PROPERTIES

In an additional area, you can find parameters depending on the chart type, e.g. bullish and bearish color settings.

STYLES AND TEMPLATES

Tradesignal offers two functions for saving and re-using settings for the contents and display of charts.

- Use styles to save the optical settings only.
- Use templates to save optical settings as well as indicators, strategies and history length. Even the symbol is saved too, but only applied to empty charts.

APPLY STYLES AND TEMPLATES

To apply a style, click on the **Styles** button in the toolbar to open a list of predefined styles. Here you can find the predefined styles as well as your own styles. Select a style.

If the default settings of chart tools were changed in the properties and saved for a style, applying this style changes the default properties of these tools. However, this only lasts while the style is applied to that chart.

When using the chart tools in a new chart, the usual user-defined settings are applied.

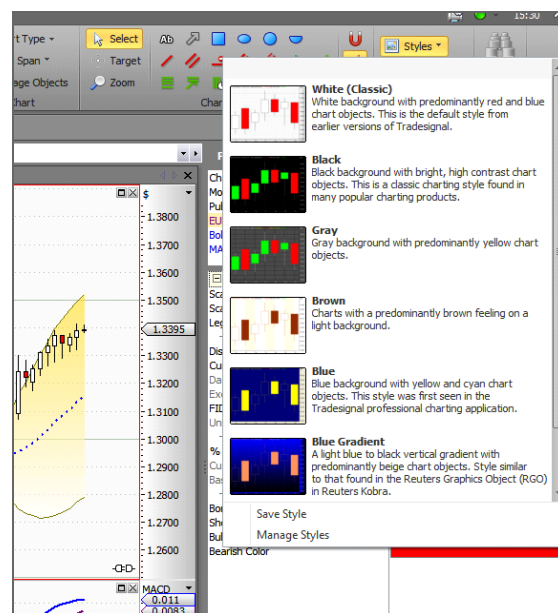
To apply a template, click on the **Template** button in the toolbox. Then open the context menu of a template and choose **Apply**.

SAVE STYLES AND TEMPLATES

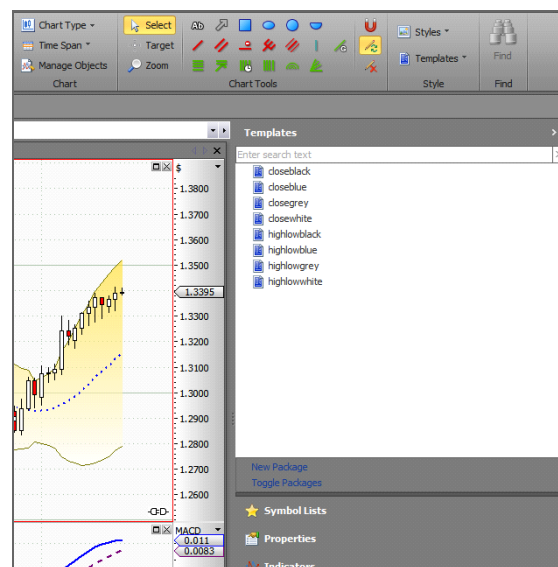
To save a style, click the **Styles** button to open the button menu and select **Save Style** from the list. A dialog opens in which you can enter the name and description of the template which will later appear in the list.

To save a template, open the context menu of the symbol chart legend and select **Save Template**. Alternatively, select **Home > Style > Templates > Save Template** on the toolbar. A dialog opens in which you have to enter a name.

For more information, see the chapter Styles and the chapter Templates.



Button menu "Styles"



Templates in the toolbox

CHART TOOLS

Chart tools are the basic tools for your chart analysis. Tradesignal offers you a wide variety of chart tools that are flexible and can be individually configured. Many of the tools offer additional features like duplicating, extending, mirroring and free positioning. You can also change the graphic display settings of tools.

Tip: The right-click context menu for a chart will always show a mini toolbar containing the available chart tool. This way, even if the Toolbar is hidden, tools are easily accessible.

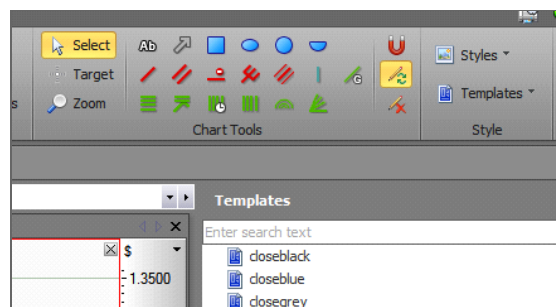





















Chart Tools

CHART TOOLS OVERVIEW

Symbol	Name	Description
	Text	Free text object in the chart.
	Fibonacci Retracement	Grid consisting of retracement and projection lines on the basis of two anchor points spaced by Fib. numbers.
	Fibonacci Extensions	Grid consisting of extension lines on the basis of three anchor points.
	Fibonacci Time Projection	Grid projection of the distance between two anchor points into the future (time axis)
	Fibonacci Time Zones	Vertical grid of line spaced by Fib. numbers.
	Fibonacci Arcs	Arc above a base line with two anchor points whose spacing corresponds to Fibonacci ratios.
	Fibonacci Fans	Fan above a base line with two anchor points whose angle spacing corresponds to Fibonacci ratios.
	Trend Line	Line between two freely settable anchor points
	Trend Channel	Trend channel between two freely settable anchor points
	Stop Line	Horizontal line for marking price levels
	Andrew's Pitchfork	Three lines (one median, one support, one resistance) according to the analysis method by Andrew.
	Linear Regression Channel	Trend channel on the basis of a regression line between two anchor points

	Gann Line	45 degree line according to the analysis method of Grad W.Gann
	Arrow	Arrow between two anchor points (automatic labeling possible)
	Guide Line	Simple vertical line with date and time
	Ellipse	Ellipse with arbitrary dimensions.
	Rectangle	Rectangle with arbitrary dimensions.
	Circle	Circle with arbitrary diameter.
	Arc	Semi-circle with arbitrary dimensions.

WORKING WITH CHART TOOLS

INSERTING A DRAWING OBJECT

To use a chart tool for entering a drawing or text object, click the respective icon. Then click into the chart and enter the object. The detailed proceeding depends on the tool: for a line, you have to enter two anchor points, for a circle you only have to click into the chart once and then drag the cursor to draw the circle.

OPENING THE CONTEXT MENU

To open the context menu of a drawing object, right-click the object.

MOVING AND SHAPING DRAWING OBJECTS

Drawing objects can be moved freely and shaped in the chart, as long as the snap mode isn't active.

- To move whole objects, click on the object. A small hand or, depending on the object, a double arrow appears. Keep the left mouse button pressed and drag and drop the object.
- To change the shape of a drawing object, e.g. for moving a single anchor point or for changing the dimensions of a circle, click the anchor or boundary point, keep the mouse button pressed, and move the point.

In the case of trend lines, trend channels and Fibonacci retracements, you can also extend the line into the moving direction by double-clicking one of the anchors points. Alternatively, you can also choose **Extend left** or **Extend right** from the context menu. The extended line ends at the edge of the chart. No additional anchor points are inserted.

DUPLICATING OBJECTS

You can duplicate every drawing or text object by double-clicking the object or by choosing **Duplicate** from the context menu. Upon duplication, all graphic properties like color, opacity, style, entered text or other specific settings are copied 1:1.

In the case of trend lines and channels, three additional duplication methods are available:

- When you press the **CTRL** key while double-clicking the object, the object is duplicated and the resulting copy is mirrored on the y-axis. This way, the moving direction is inverted.
- When you press both the **CTRL** and **Shift** keys while double-clicking the object, the object is duplicated and the resulting copy is mirrored on the x-axis.
- When you press the **SHIFT** key while double-clicking the object, the object is duplicated and the resulting copy is moved to the end point of the original object in moving direction. This way, lines and channels of the same direction and angle can be lined up with the first object.

DELETING DRAWING OBJECTS

To delete a single drawing or text object, select the object (in select mode) and press the Del key. Alternatively you can choose **Delete** from the context menu. If there are several objects under the mouse cursor, a selection list will be displayed. There will be no further confirmation dialog.

DELETING TOOLS FROM THE MANAGE OBJECTS DIALOG

To delete all drawing and text objects in the chart, click the icon



in the chart tools group, you will be shown then **Manage Objects** dialog where all tools in the chart can be selected or removed.

The **Manage Objects** dialog also provides functions to perform bulk remove operations based on the selected tool in the dialog:

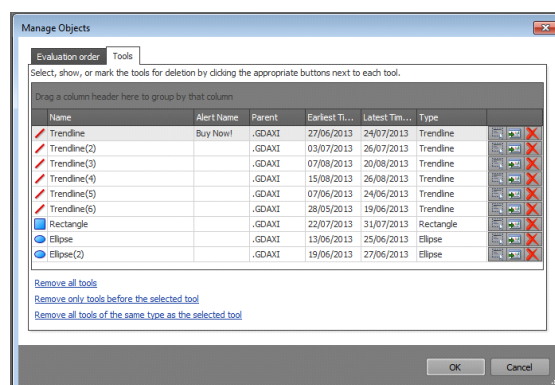
- Remove all tools
- Remove only tools before the selected tool
- Remove all tools of the same type as the selected tool

Multiple tools in the **Manage Objects** dialog can be selected at once by holding the **Ctrl** key when clicking on them. Selecting the **Delete** option with multiple selected items will delete them all.

Items will only be deleted when the **OK** button is clicked, if a tool is not desired to be removed, either click the **Undo** button next to a tool marked to be deleted, or click **Cancel** to abort all changes.

SELECTING SPECIFIC TOOLS IN A CHART

The tools tab of the **Manage Objects** dialog shows all tools in the selected chart. This list can be sorted, grouped and filtered as desired in order to locate the required tools.



Removing tools with the Manage Objects dialog

Once an item of interest has been located, clicking on the **Select the tool** button next to the item will select this tool in the chart (the property inspector will display the items properties). Selecting the **Scroll to tool** button will scroll the chart to the location of the tool.

To sort the tools by a specific column, click on the column's header (click a second time for a reverse sort).

To group by a specific column, drag the column header to the area above the list (drag the column away from this area to remove the grouping)

To filter the list, hover the mouse over a column header, and click on the little filter icon that appears in the top right of the header. A list of possible filters will be displayed and can be freely selected.

LEAVING THE DRAWING MODE

You can leave the chart tool mode by

- right-clicking within the chart
- by clicking the select tool in the chart tools group.

Note that with the icon



(Auto reset) in the chart tools group, you set a configuration so that after each chart tool use, the application automatically reverts into the select mode.

SETTINGS AND PROPERTIES

SNAP MODE

Click the magnet icon



to activate the snap mode. In the snap mode, the anchor points are drawn to the closest high or low point for exact positioning.

Example: If you set the first anchor point of a trend line close to a low point and then move the second anchor point across the chart, it will move along the other low points.

AUTO RESET

The standard setting is that once you selected a chart tool, you can enter as many objects into the chart as you want. This is the optimal setting for e.g. entering a multiple line chart analysis.

However, if you don't want to do that, you can use the auto reset mode. To set this option, click the



icon. Upon auto reset, the application automatically reverts into select mode after each use of a chart tool. To add another drawing object, you then have to select a chart tool again.

Click the



icon again to leave the auto reset mode.

DISPLAY PREFERENCES

In the property inspector you can edit various display settings for the objects. To change these settings, select the object and then click the respective property.

ALERTS AND DRAWING OBJECTS

You can assign alerts to drawing objects. You can find more about this option in the chapter Alerts.

SAVE YOUR SETTINGS AS STANDARD

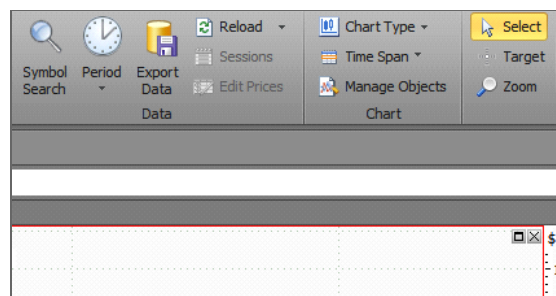
To save your personal settings as the new default for a tool, click the disc icon at the top of the property inspector and confirm the prompt with OK.

The next time you use a chart tool, your saved settings will be used.

TIMESPANS AND PERIODS

For analyzing stocks and securities, not only the navigational features of the chart but also other settings are important, like the period and timespan settings. Change them to:

- Analyze price movements and trends over various time resolutions
- Detect trends over longer time periods (days, months, years)
- Compare the performance of normalized timespans
- Optimize the chart display according to your preferences
- Change the signal generation frequency of a trading system
- Adapt the visible data to the analysis horizon



Timespan and Period icons in the toolbar

You can set the trading period not only in the chart but also in the Portfolio, the Scanner and the Watchlist. All paragraphs in this text apply analogously.

CHANGE THE PERIOD

You can change the period for all instruments in an active chart in two ways:

- Via the button **Period** under *Home > Data* on the Toolbar. Click on the button to open the button menu and select a period.
- Via the command line by entering certain shortcuts, for example "w" for week or "5h" for five hours. You can find the list of all shortcuts in the chapter Command Line.

If you want to change the period for all instruments in all charts press **Ctrl** while selecting a period.

To change the period of a single instrument in a chart, right-click on the chart legend of the instrument and change the **Period for Object** in the context menu.



Button menu Period

A trading period is the unit that corresponds to a price in the plot. Trading periods are necessary for the financial market to be able to compare analysis methods, trading systems and price patterns independently of the actual value or day of time.

KINDS OF TRADING PERIODS

Trading periods come in three types:

- Periods based on price fixing - in this case, the price is plotted for a defined number of ticks. This kind of period results in a non-linear time scale in the chart. Three periods are available: 1, 10 or 25 ticks.
- Time-dependent period - In this case, time units are used, like hour, day, month and year. Several standard time units are

offered as quick settings.

- Periods based on traded shares - In this case, a defined number of shares are accumulated to a trading period for this security. For example, you can set that for each 1000 shares, a price should be plotted. This kind of period results in a non-linear time scale in the chart. Several quick settings are available, from 1000 to 50000 shares.

DEFINE YOUR OWN TRADING PERIOD

You can define your own trading period by selecting the entry **Other Period** in the period button menu.

As a trading period, you can set every number divisible by 1 as a time and calendar unit, or set an arbitrary number for ticks and shares.

CHANGE THE TIMESPAN

For an active chart, you can find the timespan button in the *Chart* group in the Toolbar. Click on the small arrow to open the timespan menu. Select a setting.

If you want to change the timespan for all instruments in all charts press Ctrl while selecting a timespan.

The timespan is the span visible in the chart history without scrolling. You can use given settings or define your own timespans.

KINDS OF TIMESPANS

Several settings are available for the timespan:

Current (Minute, Hour...) - Display only the current minute, hour etc.

Recent History (F8, default) - In all chart types in which trading periods are linear with time, the "Recent History" is a length of 63 past trading periods and 7 empty periods of free space to the right of the current period. In charts types with a non-linear time axes, the view is adapted to what viewers might perceive as optimal arrangement.

Full History (Ctrl + F8) - Show the data as set in **History Length** in the property inspector (default 1000, maximum 500000).

DEFINE YOUR OWN TIMESPANS

To set exact timespans, call up **Fixed Date Range** or **Sliding Date Range** in the timespan button menu.

Fixed Date Range - Here you can define a start point for a timespan of a fixed length. Pre-set values are the current time and date. Click the arrow next to the **Date from** field to open a calendar to choose a date from. For the timespan, any number divisible by 1



Button menu Timespan

is possible. By default, the timespan is displayed from the start point into the future. Select **Span backwards from specified date** to display the timespan from the start point into the past.

Sliding Date Range - Here you can define a timespan of a fixed length that constantly begins with the last price of the current trading period. For the timespan, any number divisible by 1 is possible.

With the key **F8** or a click on the axes symbol on the lower right of the chart you return to the standard setting **Recent History**.

USING TEMPLATES

In templates you can save all graphic settings of a chart as well as any included indicators, strategies, and the history length as a template for future charts.

Templates are handled like documents in the toolbox. Therefore, functions like package management and color tagging are available.

SAVE A TEMPLATE

To save a template, a chart has to be active.

1. Select **Save Template** in the context menu of the chart legend. Alternatively, click on *Home > Style > Template > Save Template* in the Toolbar.
2. Enter a name.
3. Either select **Create in** to choose an existing package for saving the template, or set up a new one.
4. Confirm with **OK** to save the template.

You can find all available templates in the toolbox under the button **Templates**.



Template manager in the toolbox

APPLY A TEMPLATE

1. Click the button **Templates** in the toolbox.
2. Drag and drop a template into a selected chart or double click on it. All previously saved settings (including the securities, parameters of indicators and strategies, and the history length) are displayed in the chart. Alternatively, choose **Apply** from the context menu of the template.

If you apply a template to an existing chart, the symbols of the chart are used instead of the symbols saved in the template.

Once a template has been applied to a chart, the chart is linked to that template. Any changes to the template will be automatically applied to the chart. The template used by a chart can be seen in the property inspector.

To detach a chart from its template select the chart and then click on *Home > Style > Templates > Detach Template* on the Toolbar.

To temporarily disable the automatic updating of charts when the template changes, uncheck the box in the options under *File > Options > Advanced > Templates*.

MANAGE TEMPLATES

In the template manager in the toolbar, you can access various functions via the context menu:

- You can **apply** the template to the current chart or **open** it in a new chart.
- You can **rename** or **delete** templates.
- You can define a template as default with **Set Default**. The default template is then applied to all new charts. To return to the Tradesignal default, select **Clear Default**.
- You can use color tags on templates.

STYLES

With styles, you can edit the graphic properties of Chart, Scanner, Watchlist, Portfolio, Market Depth, and News.

You can do the following:

- assign existing styles
- save current settings as a new custom style
- edit styles in style management

APPLY STYLE

When a document of a type listed above is open, you can find the *Style* area in the Toolbar.

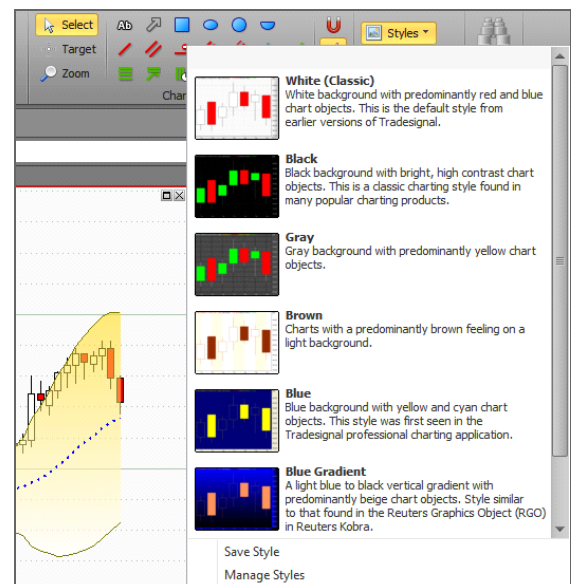
Click on the **Styles** button to open the menu.

- To apply a style to a single document, simply click on it in the list.
- To apply a style to all currently open documents (in all workspaces), press the **Ctrl** key while clicking on the style.

The style is applied immediately.

SAVE STYLE

1. Open the **Styles** button menu and click the entry **Save Style**. A dialog opens.
2. Enter a name for the style.
3. Enter a description (optional). This is the text that will later appear in the Styles button menu.
4. You can choose whether this style should be used as standard for this document type. In addition, you can choose whether the style should be **applied to all open documents of this type**.
5. Save the style by clicking **OK**.



Styles button menu in the toolbar

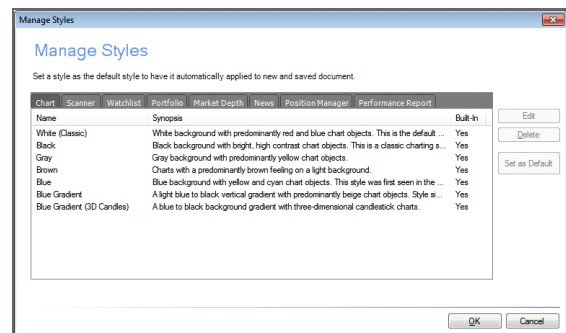
MANAGE STYLES

To manage styles, open the **Styles** button menu and choose the entry **Manage Styles**. A dialog opens.

Open the tab of the document type for which you want to edit the styles.

Click the **Set as Default** button to use the selected style as new standard for documents of that type. If you click **Clear Default**, the Tradesignal standard is used again.

If you have saved your own styles, you can **edit** them (only name and description) and **delete** them.



Style Management

FORWARD CURVE

A Forward Curve in Tradesignal allows commodity traders to obtain a visualization of the forward or futures prices on a given date for a given commodity.

The Forward Curve plots the current price of a given forward contract against the expiration date of the contract. The current prices or prices from any previous day may be plotted in such a way.

Forward Curves are available in Tradesignal, making them usable in the same way as other charts, i.e. you can apply indicators and strategies to them.



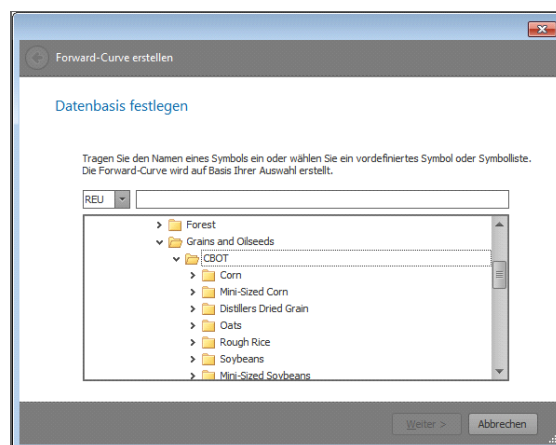
Forward Curve in Tradesignal

FORWARD CURVES IN TRADESIGNAL

CREATING A FORWARD CURVE CHART

You can access the Forward Curve Wizard from the Toolbar by clicking on the **Forward Curve** button in the **Insert** tab. It allows you to select symbol lists containing commodities or other futures and set the **Date alignment** and **Displacement** properties.

You can also drag a symbol list of commodities onto a chart, this will prompt you if you want to create a Forward Curve or insert all symbols contained in the list into the chart. If you enter the symbol shortcut for a continuous contract, e.g. "Sc1" for a contract series for soy beans, into the command line, you can select the option **New Forward Curve** from the drop-down menu.



Forward Curve Wizard

You can also create a forward curve from a user-defined list of forward contract symbols. Please be aware that for this feature to work, it will be necessary to manually mark existing user-defined lists as supporting forward curves using the **Allow Forward Curves** option in the list's context menu.

If you want to display the names of each contract as a label near each price point, you can apply the supplied **Forward Curve Label** indicator.

PROPERTIES OF A FORWARD CURVE CHART

You can edit the properties of a Forward Curve chart either during its creation in the wizard or use the property inspector in the Toolbox.

There are two properties specific to Forward Curves:

- **Date Alignment** - The expiry date of each contract will be aligned according to this setting, useful if you want to align multiple different forward curves.
- **Displacement** - The displacement property governs what value to show for each contract in the forward curve. A value of 0 means *use the current value*, 1 means *use the value of the previous bar*, etc.

PERCENT PERFORMANCE

A percent performance chart offers an easy way to compare the performance of multiple symbols.

It shows the percentage increase or decrease of a symbol relative to its price at a given **start date**, which may be selected globally in Chart Properties in the section Global Percent Performance Properties or for each symbol in the Symbol Percent Performance Properties.

You can create a percent performance chart either by setting the properties on an existing chart or by using the percent performance wizard in the toolbar to create a new chart.

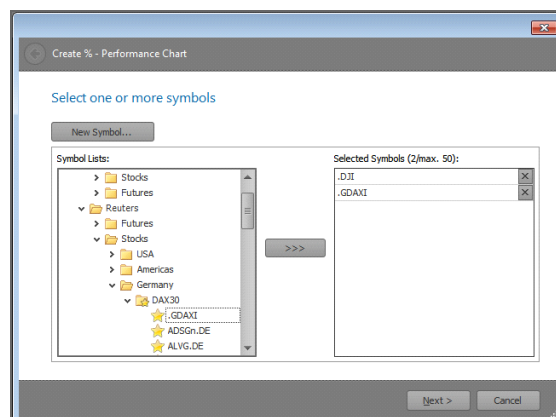


Percent performance chart

PERCENT PERFORMANCE WIZARD

The Percent Performance Wizards allows you to select up to 50 symbols from your symbol lists and to choose between several chart styles for your new chart.

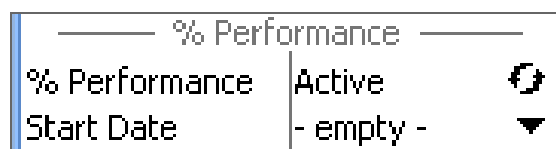
You can also set the global **start date** in the wizard. If you do not select a **start date**, the first available price will be used.



Percent performance wizard

Found in the property inspector of the chart this group of properties allows you to enable/disable the percent performance transformation for all symbols in your chart and to select a new start date.

- **% Performance** - Can be set to **Active** or **Inactive**.
- **Start Date** - The price at this date will be used as a base value from which the percent increase/decrease is calculated. If none is selected the first price is used.



Global properties

SYMBOL PERCENT PERFORMANCE PROPERTIES

The properties in the group *% performance* of the symbol properties apply only to the selected symbol.

- **% Performance - Default** uses the global setting, **On/Off** are always on/off ignoring the global setting, **On (Custom Start Date)** uses the **Start Date** below.
- **Custom Start Date** - The price at this date will be used as a base value from which the percent increase/decrease is calculated if in **On (Custom Start Date)**. Otherwise the global **Start Date** is used.
- **Base Value** - Select which price field (close, open, ...) you want to use.

% Performance		
% Performance	Default	▼
Custom Start D...	- empty -	▼
Base Value	Close	▼

Symbol properties

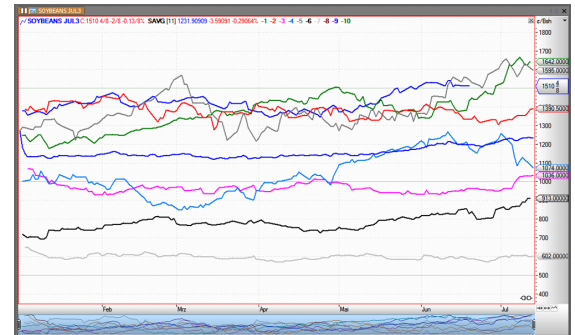
SEASONAL CHART

With the seasonal mode, Tradesignal offers you the ability to overlay the prices of different years for one or more symbols. Comparing the symbols this way emphasizes seasonal pattern, e.g. if used with commodities that have seasonal variations.

PROPERTIES OF A SEASONAL CHART

If enabled, the seasonal properties group will be displayed in the chart properties, allowing you to customize the seasonal view:

- **End Month** - Specify the end month on the seasonal date scale.
- **End Day** - Specify which day of the month should be the final day on the seasonal date scale.
- **Preceding Months** - Specify how many preceding months until the end month should be displayed on the seasonal date scale.
- **Displacement** - Use this setting to display the symbols displaced, so that they align to the selected end day with/without any weekend gaps closed.



A Seasonal Chart

SEASONAL CHART WIZARD

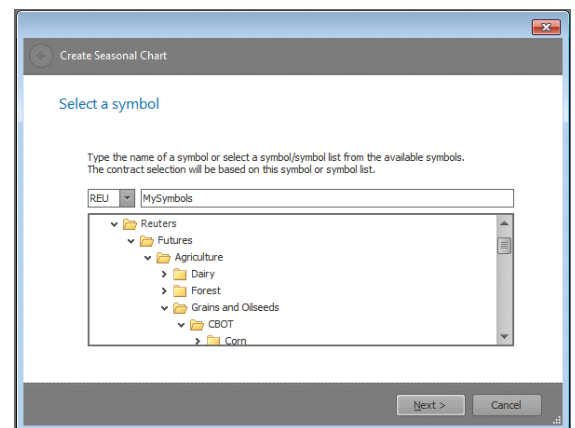
The Seasonal Chart Wizard in Tradesignal, which is found in the Toolbar, allows you to select a list of futures and to display them in a seasonal chart with the option of adding a Seasonal Average Indicator which shows the average price across all contracts in the contract series.

EXAMPLE

Example on how to compare expired wheat future contracts in seasonal mode:

1. Open a chart with the expired December wheat contract from 2007, **W 2007Z**.
2. Add a previous year contract which also expired in December, e.g. **W 2006Z**.
3. Select the chart. In the chart properties of the date axis, change the **Mode** property to **Seasonal**.

The symbols will now be displayed below each other.



Seasonal Chart Wizard

TRADING SYSTEMS AND ORDER ROUTING

STATISTICS

TOTAL COMMISSION AND SLIPPAGE

Here you can find detailed information on the commission and slippage key figures in the Performance Report.

TOTAL COMMISSION

This sum is an important, but often overlooked key figure. It shows you how much money you spent only for the trading activities. Usually, this sum has to be set in relation to the system profit. The ratio should lie far on the profitable side.

TOTAL SLIPPAGE

Usually, the slippage is seen as an uncertainty factor. Between signal generation, order entry and order filling there is a time in which the price moves, possibly against you.

To see the effects of such price movements in the backtest, the slippage is subtracted from the profit or is added to the loss. The sum shows you how much smaller the realized profit would be in comparison to the maximum in the backtest.

Here you can find detailed information on the drawdown key figures in the Performance Report.

A phase of declining profit is called a drawdown phase. The key figure Max. Drawdown displays the highest absolute profit drawdown in the time range of the test. The Performance Report also shows the date on which that max. drawdown occurred.

Performance Report (Two Time Periods)						
Statistics	Current Period	All Traders	Currency	Long Trades	Short Trades	
Total	Being open, being long	1,577		2,834		
Longest Winning Trade	1,961,000.00 USD	1,961,000.00 USD	912,000.00	1,969,000.00		
Longest Losing Trade	-395,000.00 USD	-395,000.00 USD	-395,000.00			
Max. Consecutive Winning Trades	3	3		3		
Max. Consecutive Losing Trades	6	6		6		
Average Win % Total Trades	12.891%	12.891%				
WIN % AMERICAN - LAST DAY						
Average Win % In Trade	246	246				
RODING LOSS - LAST DAY	22,884.77	24,430.67	15,668.67			
CATSTRAP LOSS - LAST DAY	7,837.04	8,742.99	8,482.67			
OVERHOLD LOSS - LAST DAY						
Average Win % Even Trades	0.0000%	0.0000%	0.0000%			
CODOL COULD - LAST DAY	0.0000%	0.0000%	0.0000%			
CODOLA COULD - LAST DAY						
Number of Consecutive Losing Trades	1	1				
Number of Consecutive Winning Trades	1	1				
IS UNL AMERICAN - LAST DAY						
Max. Share/Contracts Held	800	800				
Total Share/Contracts Held	40000	42000	42000			
DONCH RISE - LAST DAY						
Total Commission	0.0000%	0.0000%	0.0000%			
Total Storage	0.0000%	0.0000%	0.0000%			
ACCOUNT BECAME - LAST DAY	2,642,000.00 USD	1,118,000.00	2,646,000.00			
RECEIVED FUNDING - LAST DAY						
NAME DEPOSIT INCOME - LAST DAY	Return On Account	4.93%	63.5%	-12.07%		
NAME DEPOSIT - LAST DAY	Profit/Loss Trade	3,368.88				
DEPOSIT - LAST DAY	Start Date	7/23/2020				
DEPOSIT - LAST DAY	End Date	9/6/2020				
DEPOSIT - LAST DAY	Trading Period	1408 Days				
DEPOSIT - LAST DAY	Time in the Market	221.02%				
DEPOSIT - LAST DAY	Percent of Time in the Market	12.1012%				
Max. Drawdown	-2,642,000.00 USD	-1,118,000.00 USD	-2,646,000.00			
Max. Drawdown Date	5/12/2020	5/12/2020	6/9/2020			
Max. Intraday Drawdown	-5,017,000.00 USD	-4,686,000.00 USD	-3,126,000.00			
Total Profits						
Profit/Loss Changes	462	275	217			
Total Closed Positions	36	39	20			

The drawdown is calculated on basis of closed positions. Therefore, it does not reflect that for held positions, much larger drawdowns might take place.

The key figure is calculated on basis of the complete profit, including open positions. Therefore, it includes all price movements that cause a drawdown for open positions, although these drawdowns might be reduced or neutral once the position is closed.

Version 7.0

FROEHLICH FACTOR

The Froehlich Factor (FF) is a complex key figure that is supposed to measure the quality of the trading system results.

Using the Froehlich Factor, the changes in quality of a system can be investigated. This key figure also takes into account that rare extreme values can distort the system performance. Because of this, the biggest deviations from the standard are used as divisors in the calculation.

The formula of the FF is calculated as follows:

Statistics	All Trades	Current	Long Trades	Short Trades
Global				
Number of Open Trades	0	0	0	0
Number of Winning Bars	291	291	291	291
Number of Losing Bars	365	47	138	138
RECOMPENSE - LAST Daily				
Number of Open Bars	0	0	0	0
Average Trade Profit	-4.69279 USD	36.79884	-45.00000	-45.00000
Average Winning Trade	204.11111 USD	214.82033	452.00000	452.00000
Average Losing Trade	-200.00000 USD	-203.14286	-279.42857	-279.42857
Ratio Avg Win / Avg Loss	0.12769	0.81441	1.74342	1.74342
Longest Winning Trade	1,265.00000 USD	612.00000	1,565.00000	1,565.00000
Longest Losing Trade	-988.00000 USD	-505.00000	-588.00000	-588.00000
Max. Consecutive Winning Trades	3	4	3	3
Max. Consecutive Losing Trades	5	2	6	6
Average Bars in Total Trades		17.89474	11.30000	11.30000
Total Bars in Trade		445	324	324
Average Bars in Winning Trades	22.06667	24.41667	17.66667	17.66667
Average Bars in Losing Trades	7.88714	6.71429	8.42857	8.42857
Average Bars in Even Trades	0.00000	0.00000	0.00000	0.00000
Number of Consecutive Winning Trades	1	1	1	1
Number of Consecutive Losing Trades	1	1	1	1
Max. Win/Loss/Draws Held	100	100	100	100
Total Shared Contracts Held	84.000	42.000	42.000	42.000
Total Commission	0.00000 USD	0.00000	0.00000	0.00000
Total Range	0.00000 USD	0.00000	0.00000	0.00000
Account Size Required	2,545.00000 USD	1,176.00000	1,686.00000	1,686.00000
Return On Account	-6.93 %	63.50 %	-22.07 %	-22.07 %
Froehlich Factor	0.36688			
Start Date	01/02/2009			
End Date	06/02/2013			
Trading Period	1420 Days			
Time in the Market	493 Days			
Percent of Time in the Market	32.96 %			

Performance Report

$$\text{Netto Profit} * \text{Percent Winning Trades} * (\text{Profit Factor} * (\text{Average Win} / \text{Average Loss}))$$

$$\text{Max. Winning Trade} + \text{Max. Losing Trade} + \text{Max. Drawdown}$$

A value of FF = 5 for intraday and of FF > 10 for EOD systems should be expected, according to Stefan Froehlich.

SHARPE RATIO

The Sharpe Ratio is a key figure for the profitability of an investment in relation to its risk. It gives a feeling for the profit/risk ratio during the trade.

The calculation is based on a reference value for a "risk-free investment", for example a bond or a fixed deposit account. On this basis, the resulting interest for the investment is calculated. When calculating the risk of a trade, the standard deviation is used as a measure of the volatility.

The Sharpe Ratio is calculated for monthly periods, like the profits and the volatility.

Statistics	All Trades	Current	Long Trades	Short Trades
Number of Winning Trades	148	120	128	
Number of Losing Trades	151	121	122	
Number of Even Trades	5	1	4	
Number of Winning Bars	12815	7654	1281	
Number of Losing Bars	9482	2940	1945	
Number of Even Bars	62	14	48	
Average Trade Net Profit	46,24463 USD	531,88355	457,83365	
Average Winning Trade	2,035,30953 USD	2,161,53938	1,796,15570	
Average Losing Trade	-1,458,15238 USD	-1,692,68944	-1,671,65263	
Ratio Avg. Win. / Avg. Loss	1,46412	1,97825	1,06848	
Largest Winning Trade	98,070,00000 USD	70,621,00000	98,070,00000	
Largest Losing Trade	70,621,00000 USD	47,581,00000	70,621,00000	
Max. Consecutive Winning Trades	19	11	23	
Max. Consecutive Losing Trades	11	14	10	
Average Bars in Winning Trades	23,23125	19,21491		
Average Bars in Losing Trades	9,40427	9,10088		
Average Bars in Even Trades	14,60000	12,00000		
Number of Consecutive Winning Trades	1	1	3	
Number of Consecutive Losing Trades	2	3	4	
Max. Shares/Contracts Held	3100	3100	3100	
Account Size Required	386,231,00000 USD	146,883,360	164,837,100	
Return On Account	15,22 %	240,91 %	452,34 %	
Sharpe Ratio	0,22877	0,22887	0,24828	
Average Sharpe Ratio	0,19786	0,08312	0,12484	
Start Date	8/12/2009			
End Date	8/26/2013			
Trading Period	1428 Days			
Max. Intraday Drawdown	-474,726,00000 USD	-177,189,00000	-388,821,00000	
Total Positions	1317	660	637	
Total Closed Positions	1286	642	644	

Performance Report

AVERAGE SHARPE RATIO

The Average Sharpe Ratio is identical to the Sharpe Ratio but calculated over the whole history length. Since it is based on longer periods, the result is more averaged than the original key figure.

PERCENT PROFITABLE

Percent Profitable is a key figure in the Performance Report. With this key figure you get an information about the percentage of profitable trades in relation to all trades. This value is an important piece of information for testing and optimizing your trading system.

However, it cannot be evaluated without context. Depending on the configuration of the trading systems, completely different hit rates can result.

- Trend-following systems usually show lower hit rates, but the profit/loss ratio is higher.
- Trading systems with higher hit rates often have a smaller profit/loss ratio, which is additionally influenced by the increased cost caused by larger trading volumes.

Statistics	All Trades	Current	Long Trades	Short Trades
Total Profit	56,473.00000 USD	240,855.00...	281,465.00...	
Profit Factor	1.206246.00000 USD	441,699.00...	407,245.00...	
Drawdown	-1,039,968.00000 USD	-550,794.00...	-488,725.00...	
Profit Factor	0.00721	0.07209	0.07209	
Drawdown Ratio	-16,868.00000 USD	-18,869.00...	-1,100.00000	
Total Number of Trades	1236	442	644	
Percent Profitable	55.45 %	55.45 %	55.45 %	
Number of Winning Trades	548	230	328	
Number of Losing Trades	733	211	412	
Number of Even Trades	5	1	4	
Number of Winning Bars	13815	7434	6381	
Number of Losing Bars	8485	2940	5545	
Number of Even Bars	62	14	48	
Average Trade Profit	45,246.00000 USD	531,000.00...	437,000.00...	
Average Trade Loss	2,005,368.00000 USD	2,161,559.00...	1,766,107.00...	
Average Losing Trade	-1,428,100.00000 USD	-1,092,800.00...	-1,671,000.00...	
Rate And Vol. Avg. Vol.	0.00000	0.00000	0.00000	
Largest Winning Trade	98,070.00000 USD	70,021.00000	98,070.00000	
Largest Losing Trade	-70,021.00000 USD	-42,561.00...	-70,021.00...	
Max. Consecutive Winning Trades	39	11	22	
Max. Consecutive Losing Trades	21	14	25	
Average Bars in Winning Trades	21,960.00	23,231.00	19,219.00	
Average Bars in Losing Trades	8,847.00	9,150.00	8,847.00	
Average Bars in Even Trades	12,400.00	14,000.00	12,400.00	
Number of Consecutive Winning Trades	1	1	5	
Number of Consecutive Losing Trades	3	3	4	
Max. Shares/Contracts Held	3300	3300	3300	
Account Size Required	290,000.00000 USD	140,881.00...	29,427.00...	
Return On Account	0.212 %	24.01 %	0.212 %	
Break Ratio	0.10377	0.23917	-0.24639	
Average Sharpe Ratio	-0.10786	0.00000	-0.10786	

Performance Report

$$\text{Percent Profitable Trades} = (\text{Number of Winning Trades} / \text{Total Number of Trades}) * 100$$

AVERAGE TRADE NET PROFIT

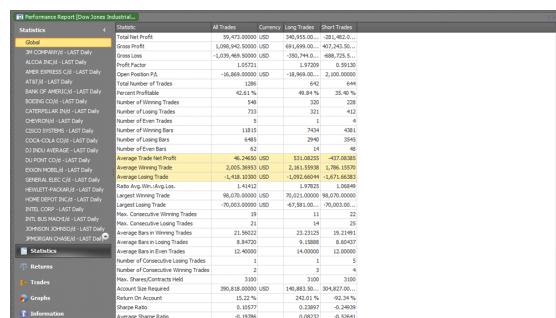
Here you can find detailed information about the Average Trade key figures in the "Tradesignal Performance Report".

In the Performance Report, all figures are displayed as cumulated valued for all trades and also for long and short trades separately.

AVERAGE TRADE NETTO PROFIT

This value gives the average profit of all trades for the data. For this, the **Total Netto Profit** is divided by the **Total Number of Trades**. The

Average Trade Netto Profit is a key figure of the trading system statistics. With its help, different results for one or several systems can be compared. The value should also be compared to the commission and slippage per trade. It helps you estimate the probability that after deducting all costs, the trading system will still be profitable.



Statistics	All Trades	Current	Long Trades	Short Trades
Total Net Profit	56,473.00000 USD	240,855.00...	281,462.00...	
Gross Profit	1,206,040.00000 USD	449,199.00...	407,245.00...	
Gross Loss	-1,039,566.00000 USD	-550,794.00...	-488,725.00...	
Profit Factor	1.03721	0.97209	0.99120	
Open Position %	-16,868.00000 USD	-18,869.00...	1,100.00000	
Total Number of Trades	1286	442	444	
Percent Profitable	62.6 %	68.8 %	55.8 %	
Number of Winning Trades	540	200	228	
Number of Losing Trades	733	221	412	
Number of Even Trades	5	1	4	
Number of Winning Bars	13915	7434	4281	
Number of Losing Bars	8481	2940	3545	
Number of Even Bars	62	14	40	
Average Trade Net Profit	43,898.00000 USD	545.00000	633.00000	
Average Winning Trade	2,005.36000 USD	2,161.55000	1,766.10000	
Average Losing Trade	-1,408.00000 USD	-1,092.66000	-1,671.60000	
Rate And Win. Avg. Win.	0.44122	0.97023	0.64849	
Largest Winning Trade	98,076.00000 USD	70,021.00000	98,076.00000	
Largest Losing Trade	-76,023.00000 USD	-47,581.00...	-76,023.00...	
Max. Consecutive Winning Trades	39	11	22	
Max. Consecutive Losing Trades	21	14	25	
Average Bars in Winning Trades	21.96022	23.21125	19.21491	
Average Bars in Losing Trades	8.84020	8.15088	8.84027	
Average Bars in Even Trades	12.40000	14.00000	12.40000	
Number of Consecutive Winning Trades	1	1	5	
Max. Shares/Contracts Held	3300	3300	3300	
Account Size Required	390,428.00000 USD	140,881.00...	390,427.00...	
Return On Account	15.12 %	24.01 %	42.24 %	
Drawn Ratio	0.10377	0.23897	-0.24639	
Average Sharpe Ratio	-0.12786	0.00232	-0.23451	

Performance Report

$$\text{Average Trade Netto Profit} = \text{Total Netto Profit} / \text{Total Number of Trades}$$

AVERAGE WINNING TRADE

This key figure gives the average of all profitable trades. Compare this value with the average of all losing trades to get an impression of the trading system performance.

$$\text{Average Winning Trade} = \text{Gross Profit} / \text{Number of Winning Trades}$$

AVERAGE LOSING TRADE

This key figure gives the average of all losing trades. Compare this value with the average of all winning trades to get an impression of the trading system performance.

$$\text{Average Losing Trade} = \text{Gross Loss} / \text{Number of Losing Trades}$$

RATIO AVERAGE WINNING TRADE / AVERAGE LOSING TRADE

This key figure gives the ratio of the two key figures above. It is important for judging the system performance. By combining it with further figures, e.g. for the number of loss and win trades, the potential of the system can be estimated.

LARGEST WINNING TRADE

This key figures gives the largest single profit of the data set. It is often preferable to ignore this value and to remove it from the

backtest result. This way, random results are less likely to influence your system.

LARGEST LOSING TRADE

This key figures gives the largest single loss of the data set. The explanation given above applies. It is often preferable to ignore this value.

USING TRADING STRATEGIES

A trading system is a collection of rules and functions from which trading decisions of any kind result. To this collection, rules for money and risk management are added. Modern computer software also supports the fully mechanized trade, up to the order routing to the broker. Trading systems are used to:

- Test a set of rules for future real trade
- Guarantee the smooth execution of trading decisions
- Sidestep human weaknesses like fear and insecurity
- Procure stable results based on clear rules
- Automate trades based on a complex set of rules
- Trade many different symbols or systems at the same time.

Tradesignal offers you a broad range of tools for the development, testing, and operation of complex trading systems:

- Strategies are written in Equilla and can be edited and adapted to your needs.
- Money Management with options for position sizing and risk management.
- Analysis of the possible trading system turnover in the Performance Report.
- Search for the most profitable and stable trading system configuration with the Optimizer.
- Mechanical trading with Automatic Order Routing.
- Alerts issued when critical limits (values or graphic lines) are intersected in defined directions.
- An easy start with the Trading Strategy Wizard that offers more than 300 strategies in a modular system, as well as new indicators and strategies for download from Tradesignal.

All of these tools are intended to help you configure and optimize your own systems.

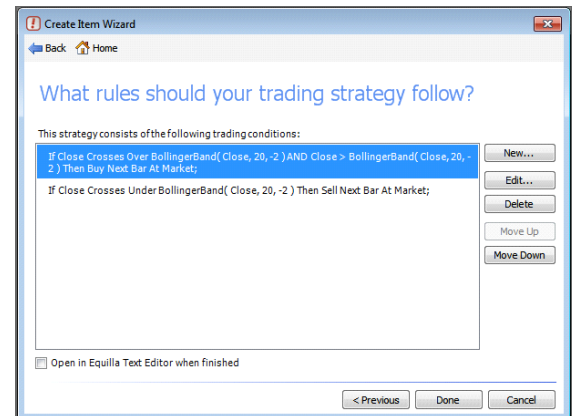
TRADING STRATEGY WIZARD

Trading strategies are an important tool for many traders who seek success at the market independent of their personal feelings, fears and insecurities. For this aim, Tradesignal offers many pre-defined indicators and strategies, which can be expanded by adjusting existing strategies or developing new strategies in Equilla.

As an alternative, Tradesignal offers a Trading Strategy Wizard to support you in assembling a strategy. Instead of scripting, simply combine lists, conditions, expressions and patterns. The wizard helps you to

- swiftly create trading strategies
- modify your strategies and customize it

Strategies created in the wizard can be used in the Chart, Scanner, Watchlist, Portfolio or the Optimizer. For all adjustable parameters, the wizard automatically creates "Input Declarations" so that you can edit the parameters in the property inspector and the Optimizer.



Trading Strategy Wizard

CREATING A TRADING STRATEGY

1. In the toolbox, click on the **Strategies** button.
2. In the *Related Tasks* area, select **New Strategy**.
3. In the next dialog, select the option **Use the Trading Strategy Wizard**.

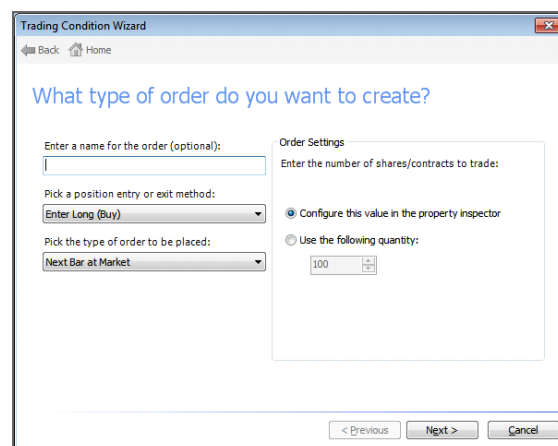
In the following window, the strategy rules are defined. Rules can be assigned to each trading order, and several trading orders can be set up for the strategy. For condition details, see below.

1. Click **New** to enter a new order.
2. Enter a **name for the order** (optional), to distinguish between different entry and exit methods.
3. Select an entry or exit method.
4. Enter an order type.
5. Select **Generate reversed trading condition** to generate also the reversed order (for example, for a "buy long" generate a "sell long" with reversed conditions).
6. Enter the Order Settings and click on **Next**.
7. In the next window, enter the conditions.

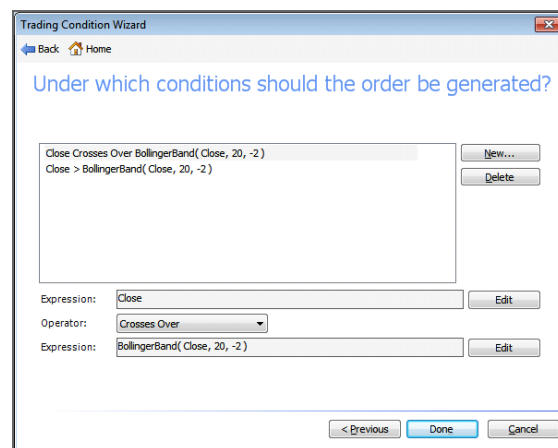
You can now edit the conditions further by clicking on **Edit** next to the Expression fields. The *Build Expression* dialog opens. Here you can define how the value is to be calculated. Select the option **Make an input** to make this value appear as parameter in the strategy properties and the Optimizer.

You can also change the operators that link the expressions. Select one from the list.

Click the **Done** buttons to save the expressions/strategy.



The opened Trading Strategy Wizard



New Strategy

ENTRY AND EXIT METHODS

The trading methods are roughly divided into exit and entry orders.

ENTRY METHODS

The following two methods create orders that open a position.

Enter Long (Buy) - A buying order is created.

Enter Short (Sell) - A short sale order is created.

EXIT METHODS

The following two methods create orders that close an existing position.

Exit Long (Sell) - A selling order is created to close a long position (a fraction or all of it).

Exit Short (Cover) - A buying order is generated to cover a short position.

STOP LOSS, TRAILING STOP AND PROFIT TARGET

Stop Loss - An order is created that closes the current position after reaching the defined loss.

Profit Target - An order is created that closes the position after reaching the defined profit.

Break Even - An order is created that prevents a position from causing losses after having made a profit.

Trailing Stop (Amount) - An order is created that follows the position development. When the position moves with the desired trend, the order "trails" behind. When the position moves against the desired trend, the position is closed after a certain decline in profit, defined as amount.

Trailing Stop (Percent) - An order is created that follows the position development. After a defined profit target, the stop order is activated. The position is closed after a certain decline in profit, defined in percent.

TRADING TYPES

An order type has to be assigned to each order. It defines when the order is executed. For two order types, additional conditions are available to control the execution.

This Bar on Close (TBOC) - The order is submitted at the end of the trading. In the settings of the Order Routing in the Tradesignal Options, you can define at what distance to the real trading end a TBOC order is created and possibly routed. In backtests, the close price of the trading period is taken as execution price.

Next Bar at Market (NBAM) - The order is submitted at the opening of the next trading period. In backtests, the open price of the trading period is taken as execution price.

Next Bar at Price Stop (NBPS) - The order is submitted during the next trading period. For a buy order, the price has to cross the given price limit, for a short sale the price has to be undercut. This way, the position is opened when the trend moves into the preferred direction.

Next Bar at Price Limit (NBPL) - The order is submitted during the next trading period. For a buy order, the price has to undercut the given price limit, for a short sale the price has to be crossed.

This Bar at Market (TBAM) - The order is submitted during the current trading period. It is executed instantly.

ORDER SETTINGS

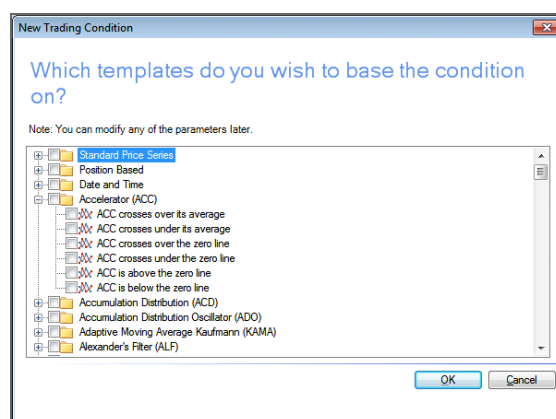
For each order, either you can define it so that the quantity of shares to be traded can be entered in the property inspector (Money Management), or you can enter a fixed quantity here.

Setting quantities in the property inspector has the benefit of making it possible to use position sizing methods that are offered in Tradesignal. This way, the quantities can be adapted dynamically in your trading system.

TRADING CONDITION TEMPLATES

The trading conditions are largely based on pre-defined Equilla scripts that include indicators, calculate chart patterns or search for candlestick patterns. For the signal generation, there are typically two expressions necessary, for example "indicator XYZ crosses over the average ABC". The calculation of the indicator and its average are all included in Equilla scripts that are combined and connected by operators in the Wizard.

The templates for signal generation are roughly listed by categories. At the top of the list, you can find three categories that generate signals based on the **Standard Price Series**, **Position based**, or based on **Date and Time**. Below these entries, you can find a long list of indicators, followed by candlestick patterns and chart patterns.



Trading Conditions

STANDARD PRICE SERIES

These signals are based on the price of the underlying stock. The signals are generated depending on comparisons of the current price to past prices. This way, changes in directions and the highest and close prices can be calculated.

POSITIONS BASED

These signals are based on the developments of existing positions.

DATE AND TIME

These signals are connected to certain dates or times. This way, you can create seasonal entries, for example. You can also create intraday strategies that are traded only within or outside of a certain time frame.

INDICATORS

This is the largest category. Here you can find all indicators included in the delivery. For each indicator, a folder with various signal variations is available. Crossings with lines, values or extremes, changes in directions and comparisons to historic values like averages are offered.

CANDLESTICK PATTERN

Here you can find a number of candlestick patterns. For each pattern, a folder with various signal variations is available. Usually, these consist of bullish and bearish patterns. Some signals are also available with trend filters.

Candlestick patterns offer one to two parameters. The first is used to define a candle body. The body is compared to the average candle body size of the given period. If a second parameter is available, it defines the period for an average that is used as trend filter. In this case, the pattern only causes a signal if it is complete and has the right position for the trend filter.

CHART PATTERN

Some common chart patterns are available here. For each pattern, a folder with a bullish and bearish pattern is available.

EDITING A GENERATED SYSTEM

WITH THE WIZARD

Any trading system generated in the Strategy Wizard can also be edited in the Strategy Wizard again. For this, right-click on the strategy in the chart legend or the toolbox to open the context menu, and select **Edit with Strategy Wizard**.

In the Strategy Wizard, click on **Edit** to edit the trading system. You can enter new conditions and rules or edit and delete old ones.

MANUALLY IN THE EDITOR

You can also open and edit a strategy in the Equilla Editor. However, please note that you will not be able to open the wizard for this strategy again if you made changes to it in the editor.

To open the trading system in the editor, right-click on the name of the strategy in the chart legend or the toolbar, and select **Edit** from the context menu. The Equilla Editor opens in which you can edit the source code.

A strategy may look as follows in the editor:

```
// Automatically generated by the Trading Strategy Wizard (1541647255)

Inputs:
CandleRawBigWhitePeriod( 5 ),
CandleRawBigBlackPeriod( 5 ),
Price1( Close ),
XAveragePeriod1( 30 ),
XAveragePeriod2( 60 );

Variables:
```

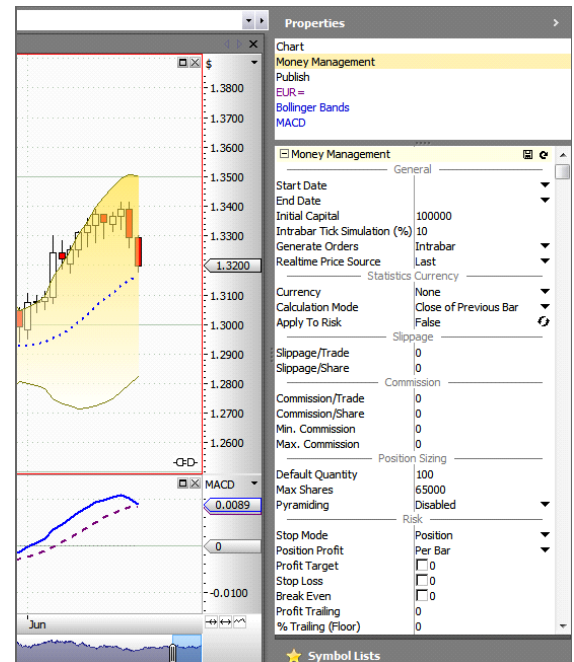
```
TRUEVALUE_VAR1,  
XAVERAGE_VAR1,  
XAVERAGE_VAR2;  
  
TRUEVALUE_VAR1 = TrueValue( );  
XAVERAGE_VAR1 = XAverage( Price1, XAveragePeriod1 );  
XAVERAGE_VAR2 = XAverage( Price1, XAveragePeriod2 );  
  
If CandleRawBigWhite( CandleRawBigWhitePeriod ) = TRUEVALUE_VAR1 Then  
Buy ( "Long Candle" ) Next Bar At Market;  
  
If CandleRawBigBlack( CandleRawBigBlackPeriod ) = TRUEVALUE_VAR1 Then  
Short ( "Short Candle" ) Next Bar At Market;  
  
If XAVERAGE_VAR1 Crosses Under XAVERAGE_VAR2 Then  
Sell ( "Long Exit" ) Next Bar At Market;  
  
If XAVERAGE_VAR1 Crosses Over XAVERAGE_VAR2 Then  
Cover Next Bar At Market;
```

MONEY MANAGEMENT

Any method for sensible capital and risk management is assembled under the concept Money Management. You can use Money Management for the following:

- Setting the initial capital of a trading system
- Setting the trading costs
- Controlling the fundamental trading system functions
- Controlling the alerts generated by the trading system
- Setting basic risk parameters

Tradesignal offers Money Management functions for Charts, Scanners, Portfolios and Watchlists. You can also read them out with Equilla code and use them in your own trading systems. This way, you always have access to information like initial capital, trading costs etc. Other functionalities that are not part of the standard settings can be set up with strategies that are partly included in the delivery.



Money Management

CONFIGURING THE MONEY MANAGEMENT

At the top of the properties window, select the **Money Management** entry to open the settings.

GENERAL

Start Date - Here you can set a date at which the trade engine should start trading. If not set, it will start trading at the first available data point.

End Date - Here you can set a date at which the trade engine should stop trading. If not set, it will continue trading indefinitely.

Initial Capital - Here you can set the initial capital for the current trading system. This is an extremely important value for calculating performance, profits, losses, etc. A system that is profitable for \$100.000 might well be a losing system at \$10.000.

Intrabar Tick Simulation % - Enter the percentage of the price span (high-low) that is used for the generation of an artificial tick in case of a stop or limit order being executed. This way, the behavior of the real price is simulated even though only the high, low, open and close values are available. The default value is "10%".

The **Generate Orders** property specifies when orders can be generated by the running strategies.

- **Intrabar** - Orders can be generated with every intra-bar update (for every tick).
- **On Bar Close** - Orders can only be generated when the bar closes. This mode is especially useful when a signal might be invalidated before the bar closes, which typically happens when the signals are generated based on compressed chart types like Point and Figure, Renko, Three Line Break and Kagi (these chart types can all remove bars resulting in trades being executed at prices that may not be reflected in a bar when it is closed).

The **Realtime Price Source** property controls which prices are used to fill orders in real time. Tradesignal supports two modes: *bid/ask* and *last*.

- **Bid/Ask** - Tradesignal uses the best bid / best ask data provided by either the broker (if an order routing module has been connected) or the data provider. If neither the broker nor the data provider is delivering bid/ask prices, Tradesignal uses the last price instead (for example in the case of indices). Best bid and best ask prices are generated when a market participant wants to buy or sell a number of shares/contracts while a last price represents an actual trade that has been made. Therefore this option has the benefit of more accurate fills for your orders. The caveat is however that the results collected in real time will not match the results of back-testing which will always only use the last price.
- **Last** - Tradesignal uses the last price delivered by the data provider. This is the same price as the one used to back-test the strategy. A last tick represents a trade that has been made in the market. When using this mode, Tradesignal can only fill and order after a trade has been made, even if there was a matching offer/request in the market prior to the trade. This mode ensures that real time results are closer to back-test results.

STATISTICS CURRENCY

When trading multiple instruments that are quoted in different currencies, aggregated statistic values can become meaningless. For example when adding the net profit of \$5 and ¥10 it is not enough to simply add the two numbers. In order to solve this both values need to be converted to the same currency before adding them. This can be either of the two given currencies or a third one. In the example both values could be converted to \$, ¥ or something else like €.

Currency - The currency global statistic values should be calculated in. The available currencies are depended on your selected data provider.

Calculation Mode - Selects the method that is used to calculate the currency conversion factors for the current bar.

- **Close of Previous Bar** - Use the currency values at the close of the previous bar.
- **Arithmetic Mean** - Calculates the arithmetic mean of Open, High, Low and Close at the current bar.
- **Midrange** - Uses the average of the highest and lowest currency value at the current bar.
- **Arithmetic Mean without Open** - The average of High, Low and Close currency values at the current bar.
- **Close Weighted Mean** - Calculate the arithmetic mean of High, Low and 2x Close at the current bar.
- **Open Weighted Mean** - Calculate the arithmetic mean of 2x Open, High and Low at the current bar.

Apply To Risk - When this option is enabled currency conversion options will also be applied to automatic stop calculations. All numeric values in the Slippage and Commission and the Risk sections are assumed to be given in the global statistics currency.

SLIPPAGE AND COMMISSION

For realistic results in the backtest and the performance evaluation of a trading system, it is important to include the trading costs in the calculation. Two parameters are available:

SLIPPAGE

The slippage is the possible difference between the price at the trading signal and the real price at order execution. Usually, some time passes between the order signal and the actual order execution, in which the price might move against the trader. You can enter the slippage per trade or share.

Slippage/Trade - Assumed Slippage per Order (independent of the number of shares), either as absolute number or as percent value (for this, enter a % character after the number).

Slippage/Share - Assumed Slippage per Shares (independent of the number of trades), either as absolute number or as percent value (for this, enter a % character after the number).

COMMISSION

The commission includes all costs of the stock exchange and broker. You can enter the commission per trade or share. In addition, you can enter a commission flat rate, if your broker offers such trading conditions.

Commission/Trade - Commission per order (independent of the number of shares), either as absolute number or as percent value (for this, enter a % character after the number).

Commission/Share - Commission per shares (independent of the number of trades), either as absolute number or as percent value (for this, enter a % character after the number).

Min. Commission - Minimum commission costs per order (if set by your broker).

Max. Commission - Maximum commission costs per order (if set by your broker).

POSITION SIZING

Default Quantity - Enter the minimum number of shares that should be traded per order.

Max Shares - Enter the maximum number of shares for this position (maximum position size). Orders that would lead to a higher position size will not get placed.

Pyramiding - Some trading systems work with position legs, meaning that after a first trading signal, newer signals will lead to further trading activities depending on the market movement. For example, the position size may be increased or decreased for a high volatility. As default, this option is inactive. If you choose pyramiding, two settings are available:

- **Yes, Different** - Further shares will be purchased in case of trading signals different to the one that caused the first action.
- **Yes, All** - Further shares will be purchased in case of trading signals of any kind.

If you select pyramiding, you can enter a maximum number of open entries (**Max. Open Entries**).

RISK

Here you can find static risk settings like Stop Loss and Trailing Stop and also more dynamic parameters. In the following, the parameters in the Money Management properties and their corresponding Equilla functions are given.

Stop Mode - Stop Mode has to be chosen in combination with the Stop Loss and Trailing Stop.

- In **Contract** mode (corresponding to **Share**), the settings of Stop Loss and Trailing Stop are interpreted in relation to each

share. For example, a Stop Loss of "100" is taken into account for each contract, independent of the position size. If the price drops by 100 from the starting value, the Stop Loss is set off.

- In **Position** mode, the settings of Stop Loss and Trailing Stop are interpreted in relation to the complete position. For example, a Stop Loss of "100" will cause a position to be closed when the total loss is 100. This might actually be four points per share for Futures or one dollar for one hundred stocks.

```
SetStopMode (Mode)
SetStopShare // Alias to SetStopContract
SetStopContract
SetStopPosition
```

Profit Target - This value in currency or points (depending on the security) gives the profit the position has to reach to be closed instantly. Profit targets may help you to stabilize your trading results over a longer time span.

```
SetStopProfitTarget (Value)
```

Stop Loss - This value in currency or points (depending on the security) gives the maximum acceptable loss after the entry, at which point the position is instantly closed. A Stop Loss can protect you from overly high losses and secure your financial ability to act in the future.

```
SetStopLoss (Value)
```

Break Even - This value in currency or points (depending on the security) gives the profit of the position that has to be reached before a position is closed once the market moves against it. It is based on the idea that positions that were once in the profit area should not move into a loss. Instead, they are closed, for example to cover the trading expenses.

```
SetStopBreakEven (Value)
```

Profit Trailing - This value in currency or points (depending on the security) gives the maximum drawback of a position profit before the position is closed. This is a simple method with which to avoid losing achieved profit.

```
SetStopProfitTrailing (Value)
```

% Trailing (Floor) - This is a more flexible version of the Profit Trailing stop. With the Floor parameter you enter a profit target at which the trailing stop will get activated. With the **% Trailing (%)** parameter you set the maximum loss (in percent) of the

position profit before the position is closed.

```
SetStopPercentTrailing(ProfitTarget, PercentLoss)
```

Include Commission/Slippage - Here you can choose whether the values for commission and slippage you entered in the above areas of the Money Management should be taken into account for the risk calculation.

ALERTS

Here you can select four possible situations (Order added, changed, cancelled, filled) in which an alert shall be issued. You can find more information in the Alerts chapter.

STRATEGIES FOR MONEY MANAGEMENT

In the scope of delivery for Tradesignal, several strategies can be found that you can use for Money Management.

STOPS AND EXITS

Fast Profit Exit - Exits a position if it yields profit the day after entry.

Peak Exit - Exits a position if it made a profit and the trade changes direction against you. It is based on a complex calculation that includes an Average True Range indicator.

Percent Trailing Stop - Corresponds to the setting in the Money Management area.

Profit Target Exit - Corresponds to the setting in the Money Management area.

Profit Trailing Stop - Corresponds to the simple Trailing Stop in the Money Management area.

Stop Loss - Corresponds to the Stop Loss setting in the Money Management area.

Timed Exit - Exits a position at a given time of day, e.g. "1900".

Timed Exit (Bars) - Exits the position at a given number of trading periods.

POSITION SIZING

Position Sizing - Fixed Fractional - Sets the position size depending on given fractions of capital.

Position Sizing - Fixed Fractional Percent - Uses a combination of the method above and other methods to define the fraction size.

Position Sizing - Fixed Percent - Uses a defined percentage of the capital for defining the position size.

Position Sizing - Fixed Risk - Measures the volatility in the market and scales the position size accordingly. Especially useful in combination with a Volatility Stop.

Position Sizing - Market's Money - Sets the position size depending on the performance of the trading system. If the total performance is in the profitable area, more positions are entered. If the performance is negative, the position size is reduced.

Position Sizing - Percent Volatility - This is a variant of the Fixed Risk method, in which position sizing is scaled depending on the volatility of the market.

MONEY MANAGEMENT AND EQUILLA

Several parameters that have to do with Money Management can be read out or set with Equilla commands. This way, you are in complete control of share numbers, open prices, commissions and slippage.

Read slippage:

```
//Gives the slippage according to the Money Management parameters
Value1 = Slippage( Price, Quantity );
```

Read commission:

```
//Gives the commission according to the Money Management parameters
Value1 = Commission( Price, Quantity );
```

Read initial capital:

```
//Gives the initial capital according to the Money Management parameters
Value1 = InitalCapital;
```

In addition, you can read out many more parameters of open and close positions. See the chapter Equilla Functions for more information.

EXAMPLE: PARAMETERS FOR A SIMPLE TRADING SYSTEM

The following parameters could be used for a trading system with Money Management based on Candlestick Engulfing Patterns:

Chart:

- **.NDX.X**
- **History Length** in the chart properties to "2000"

Strategies:

- **Candle Bearish Engulfing - Entry**
- **Candle Bullish Engulfing - Entry**

Money Management:

- **Slippage/Trade** to 2 points
- **Commission/Trade** to 2 points
- **Default Quantity** to 10 contracts
- **Stop Mode** to "Contract"
- **Profit Target** activate with a value of "225" (click on the number next to the checkbox and edit it)
- **Stop Loss** activate with a value of "40" (click on the number next to the checkbox and edit it)
- **% Trailing (Floor)** to 75 points
- **% Trailing (%)** to 5 points

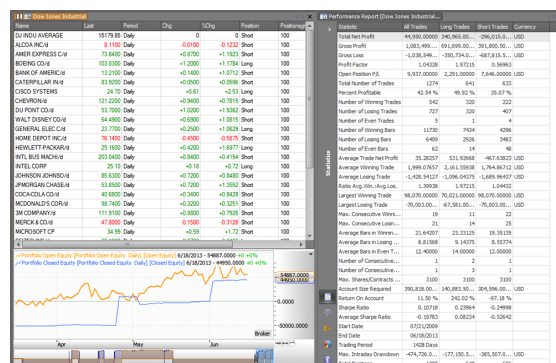
Now a first trading system is configured. You can find more strategies for Stops, Position Sizing and Statistics under **Add Strategy** in the toolbar.

PORTFOLIO

The Portfolio in Tradesignal is a basket of securities that are traded by one or more trading systems. You can use the Portfolio to do the following:

- Trade several securities in a trading system.
- Use trading systems for a list of securities
- Trade an account of several securities by defined rules for money and risk management
- Trade with trading systems automatically via a broker

Like the Watchlist, the Portfolio is updated automatically. The data is displayed in realtime. The cells with the latest changed values are displayed with colored background.



Portfolio

The important difference from Scanner and Watchlist is that the Portfolio handles the included instruments as a whole. This results in the following:

- In indicators and strategies, you can access data of other instruments in the Portfolio. (This is not possible in a Watchlist.)
- Indicators and strategies are each applied to all instruments in the Portfolio. For example, indicator 1 is applied to all instruments, then indicator 2 etc. This allows you using scripts to filter the instruments for the next steps. This is also the reason why a Portfolio cannot be displayed in a chart. (In comparison, in the Scanner, Watchlist and Chart, the indicators/strategies are applied to each instrument, i.e. indicator 1 and 2 to instrument A, then both to instrument B etc.)
- The Money Management in the Portfolio refers to the complete Portfolio. (In Scanner and Watchlist, the Money Management applies to the single instruments.)

For the idea of using scripts as filters in the Portfolio analysis, two strategies are available in Tradesignal: "Portfolio Top Dogs Picker" and "Portfolio Top Dogs Trader". First, apply the picker to pick the ten best instruments of the last year and invest into them in equal shares, then use the trader to trade these ten instruments.

The document type Portfolio has the following design parameters:

- The maximum history length (in this case of the equity chart) is 500000.
- The maximum number of instruments is 101.
- The maximum number of indicators/strategies is 32.
- Indicators and strategies are processed vertically, i.e. each is applied to all instruments before the next indicator/strategy is applied (see above).

PORTFOLIO SETUP

You can open a Portfolio with symbols in various ways.

WITH THE WIZARD

When choosing one of the following methods, the wizard will open:

- Click on the **Portfolio** button in the *Insert* tab of the toolbar.
- In the file menu, select **New** (key shortcut **Ctrl+Shift+N**) to open the Create Item Wizard. Select **Portfolio**.
- Open the context menu of a symbol list in the toolbox and choose the entry **Open in Portfolio**. (If selecting this for a single symbol, it opens without a wizard.)

The Create Item wizard opens, in which you can select indicators and strategies in two more steps. Click on **Done** to close the wizard and save the settings.

ADDING ONE OR MORE SYMBOLS MANUALLY

You can also add single symbols or open a new Portfolio without using the wizard.

- Open the context menu of a symbol or symbol list in the toolbox and choose the entry **Insert Symbol** to add it to the active Portfolio.
- In the command line, enter a symbol shortcut, e.g. "DD NYS" and select **Add Symbol** to add it to a Portfolio, or **New Portfolio** for a new one.
- Via the symbol search. For this, click on the search button and select **Insert the symbol into the selected workspace item** to add it to the Portfolio, or **Create a new Portfolio** for a new one.
- You can also add symbols to a Portfolio via drag&drop from the toolbox or other documents.

The **History Length** (candles/bars) that is regularly taken into account is 2. When you add an indicator, the length may be increased if necessary for the indicator calculation. E.g., the Chande Momentum Oscillator sets the length to 16. If you go below this necessary length, indicators may not give results anymore. In this case, increase the length again.

When you select a symbol list (e.g. the Nasdaq), please ensure that the index itself is not included in the Portfolio. It makes no sense to include an index in a list of symbols when searching for optimal trading parameters.

PORTFOLIO PROPERTIES

In the properties manager, you can edit the portfolio Properties. Important parameters are:

History Length - Enter the length of the history here (max. 500000).

Min. Referenced Bars - Enter the minimum number of bars that should be used for calculating the values of indicators and strategies.

In the *Indications* area, you can enter the colors for changing values in the table.

Update Indications – Set the background color for table cells with values changed since the last update, e.g. gray. **Up Move** and **Down Move** – Set the colors for increasing/decreasing values, e.g. green for up, red for down.

In the areas *Equity Colors* and *Equity Chart Appearance*, you can change the settings for the equity chart.

SAVING, RESTORING AND RESETTING DEFAULT SETTINGS

In the appearance area, you can find two important buttons.

- Disk button (Save as Default) - Click here to save your current settings as the new default settings.
- Restore button (Restore Default Settings, circled arrow) - Click here to reset all settings to the default settings.

Not all settings can be saved like this.

- The standard period and other user interface settings are entered in the Tradesignal Options.
- In the Portfolio, the order may be subject to changes depending on e.g. changing prices. In the advanced Tradesignal Options, parameter **Keep rows sorted in Portfolio and Watchlist (Secs.)**, you can set the time in seconds after which your original sorting order will be applied to the rows again. You can also enter default values for the history length in **Portfolio Default History Length**, for daily, intraday and tick lists.

To restore the original default settings of your Tradesignal installation, click on the **Restore Default Settings** button in the Tradesignal Options.

DELETING SYMBOLS FROM A PORTFOLIO

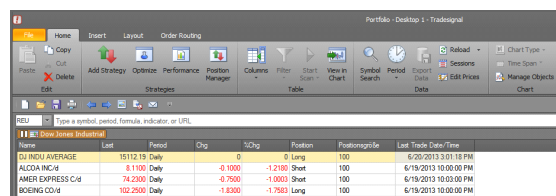
You can delete symbols from a portfolio.

1. Select the symbols. You have the following options:
 - Press **Ctrl+A** to select all.
 - Press **Ctrl** and click on single instruments to select them.
 - Press **Shift** and click on two instruments to select the instruments between them..
2. Press the **Del** key to delete the selected instruments.

PORTFOLIO BUTTONS IN THE TOOLBAR

Period - Here you can set the period that is to be used in the Portfolio (this corresponds to the period settings for a chart). The symbols in the Portfolio are always called up with the default period at first.

Columns - Open the menu for the column settings of the Portfolio.



Portfolio buttons in the toolbar

Set Group - Here you can group the parameters for better overview. Select a number of rows in the table and click on the button. You can now enter a name for the group. A button appears on top of the table that allows you to open/close the group views like a folder tree in Windows Explorer.

COLUMNS IN THE PORTFOLIO

The table shows different columns depending on the indicators and strategies you are using. The number of columns added for each indicator or strategy depends on the number of output data.

- For each indicator, all columns with a non-static output are displayed. For example, for the "Bollinger Band" indicator there are three columns, for the "Elder Ray" indicator two columns (and two invisible columns with static values).

- For strategies, the two columns "Position" and "Position Size" are displayed over all strategies. The parameters of the single strategies are usually set to invisible in the Equilla code ("visuals" are "inactive"). If they were displayed, they might also interfere with the statistics, which is usually not desired. When at least one strategy is available, you can add columns for statistical output like "Total Net Profit".

SORTING THE COLUMNS

You can sort the columns in the Portfolio by clicking on the column header. A little triangle appears, pointing up or down depending on the sorting direction. To change the direction, click on the column header again.

Alternatively, right-click into the column and select the sort direction in the context menu.

COLUMN GROUPING

This function in the *Columns* button menu leads to a new row in the table top. If groups are already defined, a button appears for them.

You can start a group by dragging a table row in this new row at the table top, or by selecting a row and clicking on **Set Group** under *Home > Table > Columns* on the toolbar.

ADD/REMOVE COLUMNS

With this function in the *Columns* button menu you can set columns in the table to visible or invisible.

In the list, all available columns are sorted by categories. Standard columns like price or symbol information are available, as well as a large numbers of statistical values for trading systems. If an indicator offers additional information, it appears as an entry here too.

Select the entries for all columns you want to display, or click on **Show All** for selecting all entries.

MANAGE STRATEGIES AND INDICATORS

With this function in the *Columns* button menu you can

- set the processing order of the indicators and strategies by selecting an entry and clicking on **Sort Ascending** or **Sort Descending**. Remember that indicators/strategies are processed vertically, i.e. each is applied to all instruments before the next indicator/strategy is applied.
- delete indicators and strategies by selecting an entry and clicking on **Remove**.

FORMAT OPTIONS

In the Portfolio, the following *Format* buttons are available in the toolbar.

Styles - Here you can set the display options of the table, e.g. gray background with orange and yellow text. You can find more information in the chapter Styles.

Sessions - Here you can set the trading times. You can find more information in the chapter Sessions, Holidays and Properties.

Price Editor - Here you can manually edit prices of symbols. You can find more information in the chapter Price Editor.

ANALYZE THE PORTFOLIO

You can apply indicators and strategies to the Portfolio. In addition, the Portfolio contents can be combined with other functions.

OPTIMIZE A STRATEGY IN A PORTFOLIO

You can start the Optimizer directly from within the Portfolio. This way, you can optimize the strategies for all symbols of the Portfolio.

Click on the button **Optimize** in the group *Strategies* of the toolbar to start the Optimizer. You have the choice between several methods, and you can choose the parameters that should be used for optimizing. The Optimizer then runs through all chosen parameters for the symbols included in the Portfolio.

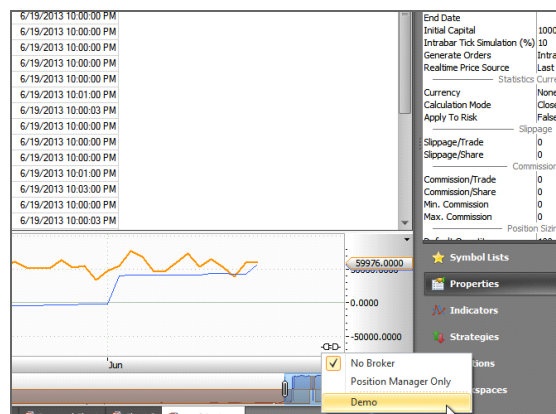
You can find more information in the chapter Optimizer.

ACTIVATE THE ORDER ROUTING FOR A PORTFOLIO

You can use Automatic Order Routing for a Portfolio. The following conditions have to be met:

- The Order Routing has to be set up correctly. See the chapter Automatic Order Routing.
- The Option **Strategy Automation Option** has to be set. For this, click on the small plug symbol in the lower right corner of the equity subchart to open the menu, and select your broker.

Afterwards, the Order Routing can be started via the entry **Start Automatic Order Routing** in the **Order Routing** tab in the toolbar.



Starting the Order Routing in the Portfolio

USE THE POSITION MANAGER WITH THE PORTFOLIO

As soon as you have assigned strategies to a Portfolio, you can also open the Position Manager for it. If not defined otherwise, the Position Manager uses the position information of the current workspace, in this case the symbols in the Portfolio. To avoid losing the overview, it is useful to use the filter function for searching information for specific symbols.

You can find more information in the chapter Position Manager.

USE THE PERFORMANCE REPORT WITH A PORTFOLIO

You can generate a Performance Report for a Portfolio. The resulting report will include performance information for the complete Portfolio as well as for each symbol.

You can find more information in the chapter Performance Report.

MONEY MANAGEMENT

The global Money Management settings can also be used for the Portfolio. Note that in the Portfolio, the Money Management is applied to all instruments as a whole. You can enter the initial capital and values for Stop Loss and Trailing Stop. It is also reasonable to use some of the Position Sizing strategies offered by the software. This allows you to distribute the available capital onto the symbols in the Portfolio as positions of the same size.

You can find more information in the chapter Money Management.

CHANGE THE HISTORY LENGTH OF THE PORTFOLIO

You can set the history length of the Portfolios in the properties of the Portfolio, setting **History Length**. Keep in mind that for long histories and large portfolios, a considerable computational effort results, which may impair your computer's free resources. Therefore, it is recommended that backtests and the real trades should be performed in separate workspaces with different history lengths.

SORT THE COLUMNS

You can sort the Portfolio columns by clicking on the respective column header. A small triangle appears (pointing up or down, depending on the sorting direction). To change the direction, click on the column header again. Alternative you can use the sorting from the context menu. Click with the right mouse button inside of the column and choose the sort direction.

OPEN A CHART FROM WITH THE PORTFOLIO AND PAGE THROUGH CHARTS

To view the chart for a symbol from within the Portfolio, open the context menu of the table row and select **Open** for the entry **Chart**. The chart is opened with the selected symbol, its period and all included indicators and strategies. The Chart will also be linked automatically to the Portfolio via symbol linking. Please note that in contrast to Scanner and Watchlist, the history length of the portfolio will not be used.

If you want to go through the whole list of symbols in a Portfolio to see each chart, use the small, green arrow buttons to the right of the Command Line. As soon as you open one chart of a Portfolio entry, you can page through the full list of symbols this way.

USING THE RESULTS WITH DRAG&DROP

USE SYMBOLS IN ANOTHER TABLE

You can reuse symbols from a Portfolio in a Watchlist, a Scanner or another Portfolio.

If you want to start with a new table, open one by clicking the button **Portfolio** (or Scanner or Watchlist) in the **Insert** tab of the toolbar and closing the wizard with **Done**. An empty table is opened. Alternatively, you can use existing tables.

In the Portfolio, click into a table cell of the symbol you want to reuse. A little plus sign and a rectangle appear under the mouse cursor. Now drag the cursor into the other table. The symbol is copied into the table.

In a similar way, you can drag symbols from a Scanner or Watchlist into a Portfolio.

SAVING SYMBOLS IN A PORTFOLIO AS SYMBOL LIST

From within the Portfolio, you can save interesting symbols as a new symbol list.

1. To save the symbols as a new list, click on the entry **New Symbol List** in the *Related Tasks* area of the symbol manager in the toolbox. Choose the option **User-Defined Symbol List** and name the list but keep it empty.
2. In the Portfolio, click into a table cell of the symbol you want to reuse. (By pressing **Shift** or **Ctrl** while clicking with the mouse, you can select more than one row.) A small plus sign and a rectangle appear below the cursor.
3. Since a first click opens the properties tab, click again on the button **Symbol Lists** in the toolbox to open the symbol manager.
4. Drag the selected symbols to the new symbol list. The symbols are added to the list.

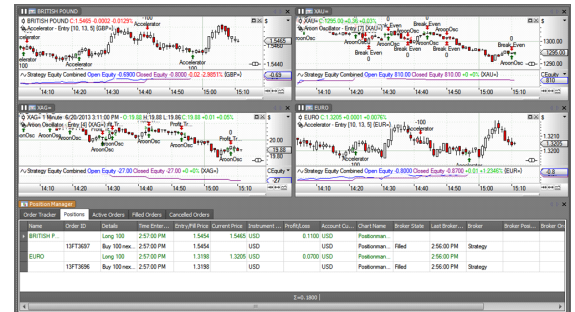
EXPORTING PORTFOLIO DATA

The table contents of the Portfolio can be copied to other programs. For more information, see the chapter Publishing.

POSITION MANAGER

The Position Manager is the central hub for trading system information. In case several trading systems are active in one or more charts, all information about held positions and pending orders etc. is displayed. This way, you have a complete overview over your current trading system activities. Use the Position Manager for the following:

- Keeping an overview of open positions
- Keeping an overview of working orders
- Keeping an overview of filled or canceled orders
- Tracking the current position development



Position Manager

Tradesignal offers you a clearly arranged Position Manager that includes all important information. It is designed as an automatically updating table with a few categories.

POSITION MANAGER SETUP

OPEN A POSITION MANAGER

In order to use the Position Manager, you have to do the following:

1. Set up a chart.
2. Set up a trading system with strategies (otherwise, no positions can be calculated).
3. Open the Position Manager, for example by clicking on the button **Position Manager** in the toolbar. The new *Position Manager* group appears.
4. Click the small plug symbol on the lower right of the chart to open the menu **Strategy Automation Options**. Choose the entry **Position Manager Only**.

Now the connection with the Position Manager is set up. You can find the current positions on the tab **Positions**.

The Position Manager collects all information from its start to its closure. No data is saved within the workspace. Therefore, when you close the workspace that includes the Position Manager, the next time you open the workspace the Position Manager starts by collecting new information.

You can set up one Position Manager for each workspace.

POSITION MANAGER BUTTONS IN THE PROPERTY INSPECTOR

Monitor All Workspaces - Usually, the Position Manager only collects information for the trading system of the currently active workspace. Click this button to collect information for all workspaces.

Show Flat Positions - Select this if positions that went to zero should also be displayed.

Font Size - Select the size of font to use in the position manager.

THE POSITION MANAGER CATEGORIES

ORDER TRACKER

In this category you can find a list of all generated orders. It lists information on the time and date of the purchase, the entry price, the profit/loss and the strategy that caused the order. In the Order Tracker you can follow all trading system activities in the workspace.

POSITIONS

In this category you can find all currently held positions. It lists information on time and date of the purchase, the entry price, current price and profit/loss, the strategy that caused the order, and the broker, in case it was bought via Automatic Order Routing. The minimum size of each position is defined by the **Default Quantity** parameters in the Money Management of the chart.

WORKING ORDERS

In this category you can find all pending orders, for example orders with a limited validity that should be filled in at a certain time or for a certain price

FILLED ORDERS

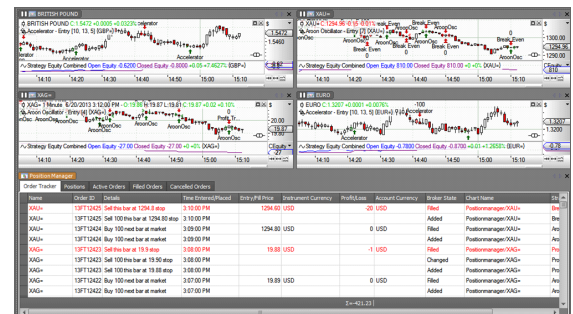
In this category you can find all orders that were filled, together with information on the time, price, quantity, the strategy that caused the order to be submitted, and the broker.

CANCELLED ORDERS

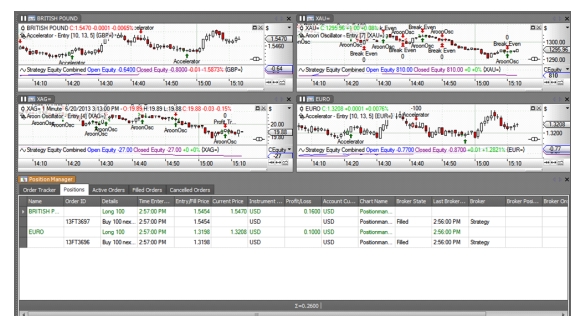
In this category you can find all orders that were cancelled before being filled. This may include Stop Orders or other order types that were cancelled by the trading system prematurely.

COLUMN SUMMARIES

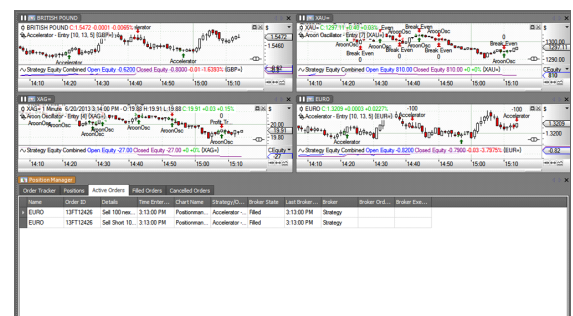
The **Positions** category supports column summaries. Right-clicking on the shaded bar at the bottom of the positions tab window will display a context menu for a specific column. This menu presents various predefined summary values that may be shown for that column (if applicable). The various summaries are



Order Tracker



Positions



Working Orders

- Sum
- Minimum Value
- Maximum Value
- Row Count
- Average Value (Mean)

Multiple summaries may be displayed for a single column.

CONDITIONAL ROW COLORING

All categories support conditional row coloring and styling. Select the category to configure and then select *Home > Table > Columns > Conditional Formatting* on the Toolbar to display the conditional formatting dialog.

1. Click the **Add new condition...** link to add a new condition and name the rule.
2. In the condition field on the right, type the condition or use the ... button to start the condition editor.
3. Select foreground and background colors and optional font changes to use if the condition is met.
4. Add more conditions if desired
5. Select **OK** to apply the formatting.

Conditions may be removed by clicking the **x** next to the conditions name in the conditional formatting dialog.

Please note: Conditional formatting will be saved with a style template for a position manager.

FILTERING RESULTS

There are multiple ways to filter the results displayed in a category in the position manager

- Quick filtering by clicking on *Home > Find > Find* on the Toolbar, and typing a keyword, matching items will be highlighted, not matching rows will be hidden.
- Show the Filter Editor by clicking on "Home > Table > Filter > Edit Filter". The filter builder allows a filter to be visually constructed using various Boolean operations and interactive components arranged in a tree structure. All following commands will also be visible in the Filter Editor if shown.
- Show the Auto Filter Row by clicking on *Home > Table > Filter > Show Auto Filter Row* on the toolbar. Only rows containing matching text entered in the filter row at the top of the column will be displayed.
- Clicking the filter icon when hovering the mouse over a column title displays a list of possible filters for the selected column.

If any filtering method other than Quick Filtering is active, a row will be displayed at the bottom of the Position Manager pane showing the active filter conditions. This row has the following options.

- A check box to temporarily disable/enable the filter.
- An **x** button to permanently remove the filter.
- A button labelled **Edit Filter** that will show the Filter Editor dialog.

ADDING, REMOVING AND REORDERING COLUMNS

Columns may be freely resized or repositioned via drag and drop. To add or remove columns, select *Home > Table > Columns > Add/Remove Columns* on the Toolbar, and drag and drop columns to and from the displayed dialog as desired.

EXPORTING POSITION MANAGER DATA

The table contents of the Position Manager can be copied to other programs. For more information, see the chapter Copying Documents via the Windows Clipboard.

EXAMPLE FOR SETTING UP THE POSITION MANAGER

With the following parameters you can follow a security with a MACD Crossover System intraday.

Chart:

- Select a security with a high turnover, e.g. "Apple Inc" (AAPL NAS)
- Set **Period** to **hourly**

Strategy:

- MACD Crossover with Bearish on Shortentry
1. In the *Insert* group in the toolbar, click on **Positions**.
 2. Click on the small plug symbol on the lower right and choose the option **Position Manager Only**.
 3. Click on the tab **Position** in the Position Manager to see the currently open positions.

OPTIMIZER

The optimizing of trading systems is one of the software's most important functions, right after developing the system logic and its programming. With an Optimizer, you can do the following:

- Evaluate the performance of a newly developed trading system
- Select the settings that give the best performance for a trading system
- Test trading systems for stability and profitability
- Generate statistical results for different parameter settings
- Configure trading systems for the usage in the Portfolio

In Tradesignal, the Optimizer is also able to optimize "in-sample" and "out-of-sample" ranges with the start and end bars defined by you. This way you can see how the trading system acts over a defined time span and if the parameter settings could also be applied successfully to other time spans.

START THE OPTIMIZER

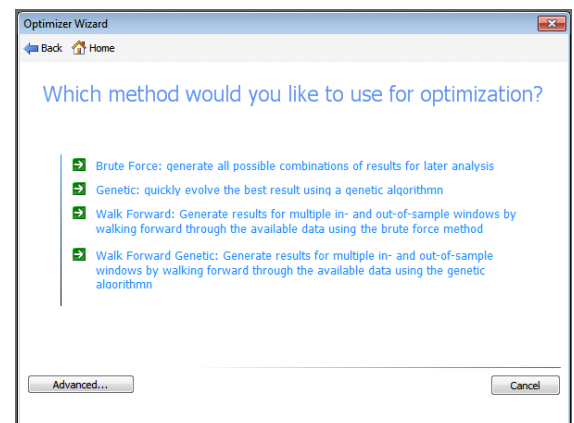
To start the Optimizer Wizard you can

- Click on the **Optimize** button in the *Strategies* area in the Toolbar or
- Select the menu entry **Optimize...** in the context menu of a symbol or symbol list in the Symbol Lists toolbox.

CALL UP THE OPTIMIZER AND CHOOSE A METHOD

The Optimizer in Tradesignal offers you several optimization methods from which you have to select the best one depending for your goals. Also note that some methods take longer to calculate the results than others.

- **Brute Force** - Depending on the parameters to be optimized and the set boundary values, the Optimizer runs through all possible combinations of parameters. This method is especially useful if you want to use the results for further analysis, e.g. in Excel. The disadvantage is the possibly high number of combinations, which may lead to a long calculation time.
- **Genetic** - Depending on the parameters to be optimized and the set boundary values, the Optimizer runs through the possible combinations of parameters with genetic algorithms that strive to find the best results. The advantage of the method is its lesser calculation time. The disadvantage is that the table of results does not include all possible combination of parameters.
- **Walk Forward** - Depending on the parameters to be optimized and the set boundary values, this method outputs complete results over defined ranges. You can define the length of the ranges. Tradesignal lists the results as "in-sample" and "out-of-sample" values. This way, you get an overview of the trading system works for smaller time samples and how stable the results are. The advantages and disadvantages are the same as for "Brute Force".



Optimizer

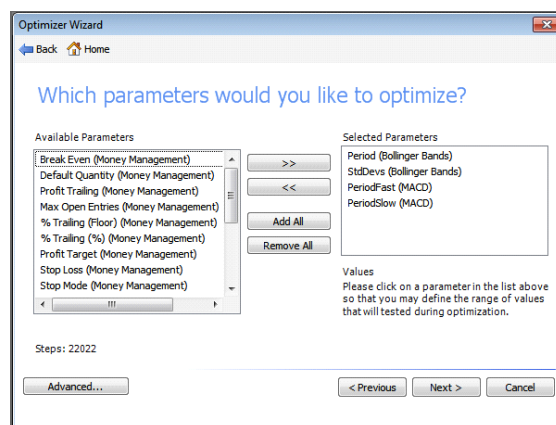
- **Walk Forward Genetic** - This method combines the defined ranges with the genetic algorithm. The advantages and disadvantages are the same as for "Genetic".

SET UP PARAMETER

After choosing the method, you have to select the parameters you want to optimize.

- In the list on the left, you can find all optimizable parameters that are available from Money Management, and the input parameters of the included trading systems. (**Note:** Parameters of indicators are also available for optimization in Tradesignal, which is useful if your trading system is based on the output of an indicator.)
- In the list on the right, the parameters to be optimized are to be found.

To move a parameter from the right to the left and vice versa, either select it and click on the respective double-arrow button, or double-click the parameter.



Optimizing Parameters

You can also select more than one parameter at the same time by holding down the **Shift** key (selecting several consecutive parameters) or the **Ctrl** key (selecting several parameters) while clicking on the parameters. To move all parameters at once from one side to the other, select **Add All** or **Remove All**.

For the selected parameters, you can enter further values, e.g. for start, end and steps. For that, select the parameter in the list on the right. The fields for the adjustable values show up at the bottom of the dialog.

(GENETIC OPTIMIZER) STATISTIC AND FINISHING CONDITION

For the genetic optimizer, you have to select which statistic should measure the success of the optimization.

- Select whether the **highest** or **lowest** value should be used.
- Select the parameter that should be used as the key parameter for the evaluation (by default **Total Net Profit**).
- Select the instrument from which the statistic should be taken. To have a list of options here, independent graphs and strategies have to be available in the current document.

In the next step, you have to set the stop point for the genetic optimization:

- **After no better result was found (in minutes)** - This can result in a long optimization duration, since you only set the time limit after finding the optimal value.
- **After a fixed period of time (in minutes)** - This allows for a very quick optimization, but might give suboptimal results.

DEFINE THE DATA RANGE

Define how much of the historic data should be used for the optimization:

- **Use all available data** - This corresponds to the **History Length** in the chart properties.

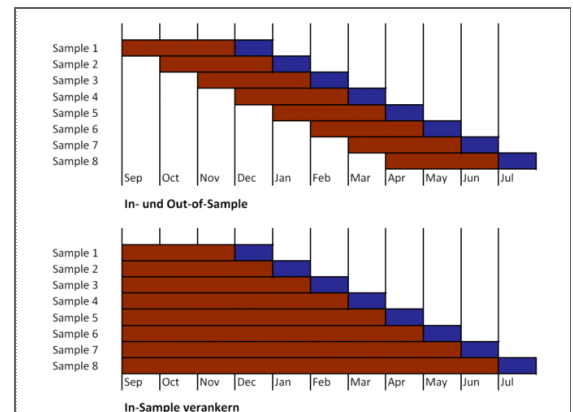
- **Use data within the following range of bars** - Set a range of data to be used. The maximum range is also given by the **History Length** in the chart properties.
- **Use data between the following dates** - Set a Start and End date for the optimization. To change the dates, either click on the calendar to select a date, or click in the area of day/month/year to change each of the values.

(WALK FORWARD) DEFINE THE SAMPLES

For the Walk Forward methods, you can define the in-samples and out-of-samples bars per sample. The method works as follows:

1. If you set in-sample to "100" and out-of-sample to "50", the optimizer will use the first 100 bars after the start point for the optimization, and then calculate the values for the next 50 bars with the results from the first 100.
2. After this, the starting point is moved forward by the range of the out-of-sample length (50 bars), and the above described optimization calculation will be executed again.

This will be repeated as long as enough data is available to fill the in- and out-of-sample ranges.



In- und Out-of-Sample

If you choose the option **Anchor in-sample: Start always at the beginning of specified data range**, the start point is not moved. Instead, the in-sample starts always at the beginning of the data range, so the in-sample increases in size over time.

START THE OPTIMIZER

Start the optimization by clicking on the **Optimize** button in the wizard.

OPTIMIZER BUTTONS IN THE TOOLBAR

In the toolbar, the *Table* group offers the following functions:

Start/Pause - Click on **Pause** to get an intermediate result of the optimization. You can start the optimizer again by clicking on **Start**.

Stop - End the optimization.

View in Chart - Apply the values selected in the table. Use the context menu on the optimizer grid to open a new chart.



Optimizer group in the toolbar

Columns - Via this button menu you can find the following options:

- **Column Grouping** - Toggles between enable and disable column grouping.
- **Add/Remove Column** - Here you can define which columns should be displayed (on the right) and which not (on the left). The indicators correspond to the ones in the Performance Report.
- **Set Graph Axes as Default** - Choose this option to use the current chart settings as default in future charts.

PROPERTY INSPECTOR SETTINGS

Runs - Here you can find a list of all runs of the optimizer with date and time. Select an entry from the list to call up the data of that run. Note that the list only displays multiple runs of a specific optimization (called up with **Start**). When you create another optimization, the counter starts anew for this optimization and another graph and table with results will be displayed.

ADVANCED OPTIMIZER SETTINGS

Via the **Advanced** button you can open a dialog with more options.

Advanced General Optimizer Options

- **Number of optimizer threads** - Enter a number higher than one, if several optimization routes should be calculated at the same time. You should change the value only if you have a computer with a multi-processors and should ideally not be higher than the amount of CPUs.
- **Thread Priority** - Select **high**, **normal** or **low**. This regulates how much system resources the Optimizer is allowed to use.
- **Use a longer history length when optimizing** - Allows you entering a history length independent of the chart properties.
- **Auto-start Excel after optimization** - If you select this option, the calculation results will be opened automatically in Excel after the optimization.

Advanced Genetic Optimizer Options

- **Only keep better results** - Only the best results are kept and displayed.
- **Show all Results** - This will keep and display all results. This may lead to a very long table.

POSSIBLE OPTIMIZATIONS

OPTIMIZING A SINGLE SECURITY

You can optimize a trading system for a single chart or symbol.

- If you have an active chart with a strategy, the **Optimize** button in the symbol bar is available. Click on it to start the Optimizer wizard.
- Alternatively, you can choose the entry **Optimize** in the context menu of a symbol.

OPTIMIZING A SYMBOL LIST

You have several choices for starting the Optimizer for a list of symbols.

FULL SYMBOL LISTS

The easiest method is optimizing a complete symbol list. To do this, choose the entry **Optimize** in the context menu of a symbol list.

LIST IN THE SCANNER OR WATCHLIST

The second way is optimizing from within a Scanner or Watchlist. When you have a list of symbols in the Scanner/Watchlist, click on the **Optimize** button in the Toolbar.

SEVERAL SYMBOLS IN A CHART

The third way is optimizing a chart that includes one or more symbols. To do this, click on the **Optimize** button in the Toolbar while the chart is active.

PORTFOLIO

Another possibility is the optimization of a Portfolio. To start optimizing a Portfolio, click on the **Optimize** button in the Toolbar while the Portfolio is active.

The difference between a Portfolio and a Scanner, Watchlist or Chart with several symbols is the combination of strategies and a single account with several symbols in the Portfolio. This way, you can trade several symbols in one Portfolio and generate global trading statistics. Therefore, you receive realistic results for the superposition of trading several symbols at various times in the Optimizer. It allows you to test various trading systems and gives you access to all statistics resulting from all possible parameter combinations.

ANALYZING AND INTERPRETING THE OPTIMIZATION RESULTS

USE THE TABLE

On the right side of the optimization window, a table with the optimization results is displayed.

- You can change the sorting order by clicking on the column header. For example, sort for the longest period of a positive trade.
- You can also group columns. For this, drag the column header into the very top of the table. This way, you can group any columns to one single group, which can be opened and closed much like the folder tree in Windows Explorer.

USE THE FILTER

Especially in large tables it is hard to keep the overview. It is also often preferable to ignore certain indicator values. To filter the view, click on the **Edit Filter** button. A simple example would be filtering for "Total Net Profit > 0".

You can find more information in the chapter Filter.

APPLY RESULTS TO OTHER DOCUMENTS

To apply optimized values to the optimized chart/portfolio, double-click the respective table row. You will be asked if you really

want to apply the values. Note that you cannot apply single cell contents, only the full row.

To use the values in a new chart or portfolio, choose the options you prefer in the button menu **View in Chart** in the Toolbar.

EXPORT OF OPTIMIZER RESULTS

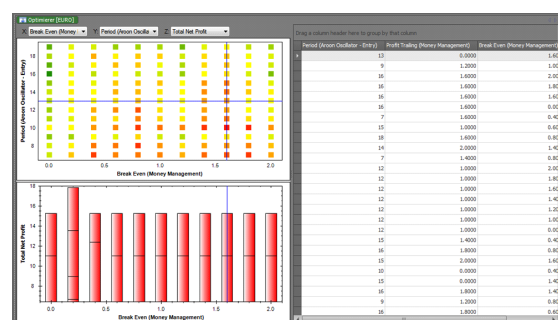
The results of the Optimizer can be exported via the file menu. Choose the menu entry **Export "Optimizer"** from the file menu. A request appears. Choose whether you want to export all runs or only the latest. The file dialog opens in which you can specify the directory, file name and file type for the export file before saving it.

GRAPHIC ANALYSIS

The interpretation of statistic data is often easier when the data is visualized in a diagram. Tradesignal offers a visualization function. As soon as the Optimizer has finished (or is paused), you can find diagrams on the left of the table. Three axes are available; values can be set via the drop-down lists. There are two kinds of diagrams:

3D (SURFACE) DIAGRAM

This is the standard setting. Choose parameters of the trading system for both the x- and y- axes, and then select a key figure like "Total Net Profit" for the z-axis.



3D Diagram

The colored surface shows the development of the key figure in relation to the parameter combination. The different colors mark different values of the key figure (green = positive, yellow = neutral, red = negative). Therefore, the diagram can be read as an elevation profile. When you click on a colored square, the corresponding row in the optimization results table is focused. In addition, you can find a diagram with the key figure and the x-axis parameter below the 3D profile.

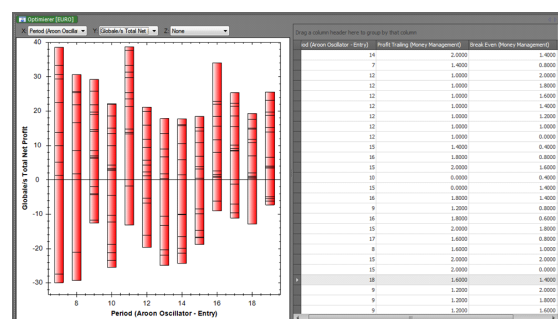
2D DIAGRAM

The simpler variant is the 2D profile. For this, select a key figure for the y-axis and one of the parameters for the x-axis. Set the z-axis to "none". This way, you can have a look at the effect of a single parameter in the trading system. The height of the columns shows the maximum values. The black lines in the columns mark the trading systems results.

The values of the tables and the diagrams are connected.

- If you click on a value in the table, cross-hairs mark the corresponding position in the diagram.
- If you click on a position in the surface or the column diagram, the corresponding value in the table is focused.

In addition, you can see the numeric values of the position beneath the mouse cursor to the right of the diagram headlines.



2D Diagram

EXAMPLE: OPTIMIZING THE STRATEGY BOLLINGER BAND BREAKOUT FOR APPLE COMPUTER, INC. (AAPL NAS)

1. Click on the **Symbol Lists** button in the toolbox.
2. Open the USA/S&P 100 [100]. (If this symbol list is not available, download this list via **New Symbol List** in the *Related Task* area.)
3. Search for the Apple Computer, Inc (AAPL NAS) symbol.
4. Right-click on the symbol and choose **Open in Chart** from the context menu.
5. In the *Trading System* group in the toolbar, click on **Add Strategy**.
6. Select **Bollinger Band Breakout** (in the tab *Complete Systems*).
7. On the tab *Position Sizing*, select **Position Sizing - Percent Volatility**.
8. Click on **OK**.



Optimizing a strategy

The selected strategies are applied to the chart with their default settings, next step is to optimize these settings to yield a higher net profit for historic values.

1. In the toolbar, click on the **Optimize** button.
2. Select the method **Genetic**.
3. Remove all select parameters from the list except **Period** and **StdDevs** (for example by double-clicking the parameters in the right list).
4. You can select different ranges and step sizes for each parameter, but in this example we use the default settings.
5. Click to the next windows. For the finishing condition, select the fixed period of time and set it to "1" minute. Start the Optimizer.

The Optimizer calculates the best results for the strategy. After one minute, the Optimizer stops and the table of results is displayed.

PERFORMANCE REPORT

The Performance Report offers statistical data from the operation or backtest of a trading system. Use the Performance Report for the following:

- Evaluating the profitability of your trading system
- Finding out if a real trade would be reasonable
- Evaluating the risks that can develop during the trade
- Estimating the need for capital
- Comparing the distribution of profits over time
- Keeping an eye on the development of real trading results

The screenshot shows a window titled 'Performance Report (Down Jones Industrial...)'. It contains a table with columns: 'Statistics', 'All Trades', 'Long Trades', 'Short Trades', and 'Currency'. The table lists various performance metrics such as Total Net Profit, Gross Profit, Gross Loss, Profit Factor, Open Position P/L, Total Number of Trades, Percent Profitable, Number of Winning Trades, Number of Losing Trades, Number of Even Trades, etc.

Performance Report

In the Performance Report, statistical results are offered as tables and diagrams.

The navigation to the right of the performance report can be toggled by clicking on the small arrow to the right of the title.

Control of which items get printed can be found in the **publish** group of the property inspector.

CALL UP THE PERFORMANCE REPORT

The Performance Report is available for Charts and Portfolios.

Click on the **Performance** button in the *Trading System* group in the Toolbar to open it. The results are calculated for the current workspace.

STATISTIC VALUES

GLOBAL

This is the default setting. In this view, you can see the results for the complete history length as a Strategy Performance Report. The results are offered in three columns. The first column shows the respective key figure for all trades, the second for long trades only and the third for short trades.

Key Figure	Explanation
Total Net Profit	Sum of all trades (profit and loss trades) after subtraction of all costs and slippages
Gross Profit	Sum of all winning trades
Gross Loss	Sum of all losing trades
Profit Factor	The ratio of Gross Profit and the negative Gross Loss. This factor is a benchmark for how large the expected profit per units of capital would have been in the test data.
Open Position P/L	Value of an open position during the analysis

Total Number of Trades	Total number of trades
Percent Profitable	Percentage of profitable trades for the total number of trades
Number of Winning Trades	Number of profitable trades
Number of Losing Trades	Number of unprofitable trades
Number of Even Trades	Number of trades closed with neutral result
Number of Winning Bars	Number of trading periods/candles/bars over which the profitable trades lasted
Number of Losing Bars	Number of trading periods/candles/bars over which the unprofitable trades lasted
Number of Even Bars	Number of trading periods/candles/bars over which the neutral trades lasted
Average Trade Net Profit	Average result of all trades, calculated as Net Profit/Total Number Trades
Average Winning Trade	Average profit of the winning trades, calculated as Gross Profit/Number Winning Trades
Average Losing Trade	Average profit of the losing trades, calculated as Gross Loss/Number Losing Trades
Ratio Average Win. / Average Los.	Ratio of Average Profit/Average Loss
Largest Winning Trade	The largest profit
Largest Losing Trade	The largest loss
Max. Consecutive Winning Trades	Longest series of profitable trades
Max. Consecutive Losing Trades	Longest series of unprofitable trades
Average Bars in Total Trades	Average number of periods per trade
Total Bars in Trade	Sum of all trading periods/candles/bars over which open positions were held
Average Bars in Winning Trades	Average duration of profitable trades in periods/candles/bars
Average Bars in Losing Trades	Average duration of unprofitable trades in periods/candles/bars
Average Bars in Even Trades	Average duration of neutral trades in periods/candles/bars
Num. of Consecutive Losing Trades	Length of the latest series of unprofitable trades
Num. of consecutive Winning Trades	Length of the latest series of profitable trades
Max. Shares / Contracts Held	Biggest single position
Total Shares / Contracts Held	Sum of all traded shares
Total Commission	Sum of all costs

Total Slippage	Sum of slippage over all trades
Account Size Required	Necessary account size to trade the system
Return on Account	Percentage development of the invested capital
Sharpe Ratio	Index for the profitability of an investment compared to the risk
Average Sharpe Ratio	Sharpe Ratio over the full data range
Froehlich Factor	Complex indicator for evaluating the quality of the trading system results
Start Date	Date of the first period
End Date	Date of the last period
Trading Period	Timespan of the data (History Length)
Time in the Market	Number of days or intraday periods in which positions were open, including weekends and holidays (absolute value)
Percent of Time in the Market	Percentage of days with open positions
Max. Drawdown	Largest cutback in profit
Max. Drawdown Date	Date of the largest cutback in profit
Max. Intraday Drawdown	Largest cutback of profit since the last open equity maximum
Total Positions	Number of held positions. If partial trades are used, higher values result than for full trades.
Position Changes	Number of switches between Long and Short trades
Total Closed Positions	Number of positions kept by the system. If partial trades are used, this key figure is higher than the Total Number of Trades.

On the left of the Performance Report, you can find further links to display trading system results as periods and graphic displays.

PERIODIC RESULTS

The returns are shown in three ways. As key figures, the Net Profit, Number of Trades and % Profitable are displayed.

- **Returns/Month** - Display the performance development for each month of the test period.
- **Returns/Quarter** - Display the performance development for each quarter (three months) of the test period.
- **Returns/Year** - Display the performance development for each year of the test period.

It is possible to show column summaries in the footer of this view by right-

Period	Date	Net Profit	Trades	Percent Profitable	Currency
Month	Jul 2009	0.000000	0	0.00 %	USD
Month	Aug 2009	-20.746.00...	1	0.00 %	USD
Month	Sep 2009	-4.171.000000	2	0.00 %	USD
Month	Oct 2009	-45.191.00...	3	0.00 %	USD
Month	Nov 2009	-28.845.00...	1	0.00 %	USD
Month	Dec 2009	2.0000000000	2	50.00 %	USD
Month	Jan 2010	-28.378.000000	1	0.00 %	USD
Month	Feb 2010	-12.610.00...	1	100.00 %	USD
Month	Mar 2010	51.137.000000	0	0.00 %	USD
Month	Apr 2010	11.188.000000	0	0.00 %	USD
Month	May 2010	84.618.000000	3	66.67 %	USD
Month	Jun 2010	-28.222.00...	6	50.00 %	USD
Month	Jul 2010	-54.453.00...	2	0.00 %	USD
Month	Aug 2010	18.137.000000	2	0.00 %	USD
Month	Sep 2010	12.336.000000	2	50.00 %	USD
Month	Oct 2010	25.153.000000	2	50.00 %	USD
Month	Nov 2010	-4.890.000000	2	100.00 %	USD
Month	Dec 2010	-14.979.00...	2	0.00 %	USD
Month	Jan 2011	31.277.000000	3	33.33 %	USD
Month	Feb 2011	371.000000	2	33.33 %	USD
Month	Mar 2011	11.030.000000	3	33.33 %	USD
Month	Apr 2011	14.801.000000	21	33.33 %	USD
Month	May 2011	-25.577.00...	24	27.50 %	USD
Month	Jun 2011	81.421.000000	42	50.00 %	USD
Month	Jul 2011	25.134.000000	55	45.45 %	USD
Month	Aug 2011	121.273.00...	48	79.17 %	USD
Month	Sep 2011	-18.194.00...	17	27.27 %	USD
Month	Oct 2011	689.000000	46	34.78 %	USD
Month	Nov 2011	189.112.00...	60	43.33 %	USD
Month	Dec 2011	-10.001.00...	47	10.64 %	USD
Summary	S=42,694.1	S=1,081	S=10.94%		

Periodic Results

clicking on a column's footer and selecting one or more of the summary calculations.

GRAPHIC RESULTS

For the graphic display, the two links offer four different diagrams:

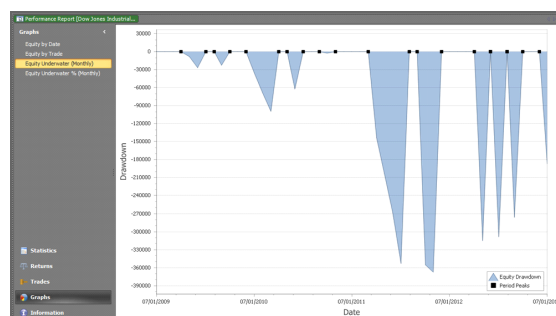
Graphs - Equity.

- **Equity by Trade** - The equities are shown in the trade sequence.
- **Equity by Date** - The equities are shown in the sequence of dates.

Graph - Equity Underwater, this means the drawdown/cutback:

- **Monthly Equity Underwater** - Phases in which the profit was declining, in absolute values of the traded currencies or units.
- **Monthly Equity Underwater (Percentage)** - Phases in which the profit was declining, in percentage values. This diagram makes it possible to compare drawdown phases over longer time spans.

The chart styles and colors can be changed by selecting the options in the property inspector.



Drawdown Analysis

DISPLAY BY TRADES

Next to the links for the statistics you can find the list of trades sorted by date (50 trades/page by default). Click on a number link to view the trades.

To return to the statistics, click on **Statistics**.

Column	Description
#	Serial number of the trade
Command	Labeling whether the trade was sell or buy, short or long.
Symbol	Traded security
Fill Date	Day of the trade
Fill Price	Exact price of the purchase
Profit	Absolute profit of the trade
Signal	Part of the trading system that set off the trade, for example entry or stop indicators
Com.	Commission as entered in Money Management
Slip.	Slippage as entered in Money Management

Display by Trades

It is possible to show column summaries in the footer of this view by right-clicking on a column's footer and selecting one or more of the summary calculations.

POSITION SIZING - VARIOUS METHODS

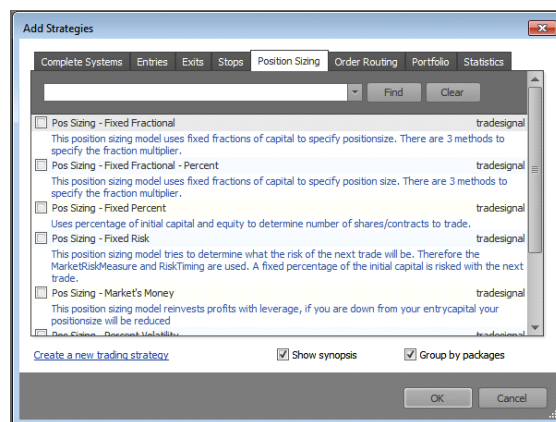
In the course of their career, many traders realize that their success at the market is less governed by entry signals, filter techniques or analysis routines. Money management, i.e. controlling the risks and the investment of capital are the crucial factors for successful trading. One of the most important aspects of the money management is the position sizing, i.e. the number of shares/contracts per position.

In the following chapter, various methods are given for position sizing.

ADDING THE POSITION SIZING STRATEGIES

1. In the *Trading Systems* area of the toolbar, click on the **Add Strategies** button.
2. Click on the tab **Position Sizing**.

Here you can select up to six strategies for position sizing to be added to your chart or portfolio.



Add Strategy

METHODS

FIXED FRACTIONAL

This position sizing model uses fixed fractions of capital to specify position size. There are four methods to specify the fraction multiplier.

FixedFracModel - Here you can choose between four models for the calculation of the fixed fractions:

- **Units** - Here you can enter a fixed amount of capital; for instance, use 5000 to trade 1 contract for every 5000 euro ($\%Toinvest = 1 / X$).
- **MaxRisk** - Here you can specify the expected biggest loss in points to use this model.
- **Capital** - This is used for future trading only, you can take the margin into account.
- **FixRisk**

FractionofCapital - Enter what fraction of capital (according to the calculation above) is to be used per trade.

BiggestLoss - For the *MaxRisk* model, enter the biggest loss in points here.

X - For the **Units** model, enter the "X" value here.

FixedRisk - When using the model *Fixed Risk*, enter the value for the fixed risk in points here.



Example for Fixed Fractional

InitialMargin - Here you can enter the initial margin. It determines how much you can invest in order to ensure your account has sufficient reserves to cover a potential margin call

MaxLeverage - Here you can enter the maximum leverage as multiplier that you have available. If you use 2x leverage, you get a loan from the broker equivalent to 100% of your investment.

Reduce - Some of the options may lead to overstepping the originally calculated investment capital or risk parameters. For such cases, you can activate this option to ensure that the position sizes are reduced to meet the criteria of the main method.

Instrument Count - Here you can enter the number of instruments in the chart. This way, a portfolio can be simulated in a single chart. It allows you to distribute your available funds across multiple instruments in one chart.

FIXED FRACTIONAL - PERCENT

This position sizing model uses fixed fractions in percent of capital to specify position size. There are four methods to specify the fraction multiplier.

FixedFracModel - Here you can choose between four models for the calculation of the fixed fractions:

- **Units** - Here you can enter a fixed amount of capital; for instance, use 5000 to trade 1 contract for every 5000 euro ($\%Toinvest = 1 / X$).
- **MaxRisk** - Here you can specify the expected biggest loss in points to use this model.
- **Capital** - This is used for future trading only, you can take the margin into account.
- **FixRisk** -



Example for Fixed Fractional - Percent

FractionOfCapital - Enter what fraction of capital (according to the calculation above) is to be used per trade.

BiggestLossPercent - For the *MaxRisk* model, enter the biggest loss in percent here.

X - For the **Units** model, enter the "X" value here.

FixedRisk - When using the model *Fixed Risk*, enter the value for the fixed risk in points here.

InitialMargin - Here you can enter the initial margin. It determines how much you can invest in order to ensure your account has sufficient reserves to cover a potential margin call.

MaxLeverage - Here you can enter the maximum leverage as multiplier that you have available. If you use 2x leverage, you get a load for the broker equivalent to 100% of your investment.

Reduce - Some of the options may lead to overstepping the originally calculated investment capital or risk parameters. For such cases, you can activate this option to ensure that the position sizes are reduced to meet the criteria of the main method.

Instrument Count - Here you can enter the number of instruments in the chart. This way, a portfolio can be simulated in a single chart. It allows you to distribute your available funds across multiple instruments in one chart.

FIXED PERCENT

This method is one of the simplest. It invests a fixed percentage of the available capital and equity in the next trade. Neither the current market situation nor any information about past trades are considered.

The example chart shows the strategy *Stochastic Momentum - Entry* (delivered with Tradesignal). The position sizing strategy invests a fixed 5% of the initial capital of 10000 USD into the first trade. The broker offers a leverage of 1:20, so that with a capital of 10000 EUR, 7784 shares at a quote of 1.2846 USD can be traded.



Example for Fixed Percent

Options

PercentToInvest - Percent of initial capital to invest in the next trade.

FixedMargin - Here you can enter a fixed margin, e.g. when trading currencies.

MaxShares - Here you can set a maximum share per position.

Reset - Here you can specify after which time you take your profits from the market and the invested capital is reset to the original initial capital. You can choose between **yearly**, **monthly**, **Never** (all capital is reinvested continuously) or **No Position Sizing**.

Instrument Count - Here you can enter the number of instruments in the chart. This way, a portfolio can be simulated in a single chart. It allows you to distribute your available funds across multiple instruments in one chart.

FIXED RISK

This position sizing model tries to determine what the risk of the next trade will be. A fixed percentage of the initial capital is risked with the next trade.

This method is especially useful when you define a stop loss for each trade. The method calculates the volatility of the base value and multiplies it with a factor set by you. This way, the position size can optimally be linked to the market conditions.



Example for Fixed Risk

The method assumes that you are using a stop loss. It is recommended that you use a volatility-driven stop. The position size is calculated so that the maximum loss per trade set by you is not exceeded.

The example chart shows the strategy *Stochastic Momentum - Entry* (delivered with Tradesignal). Once the program has measured

the volatility, the risked capital is divided by the volatility. If a leverage is used, it is included in the calculation. If you have set a fixed margin, the risked capital is divided by the fixed margin. In all other cases, the calculated risked amount is divided by the current close price of the base value.

Options

PercentToInvest - Percent of initial capital to invest in the next trade.

FixedMargin - Here you can enter a fixed margin, e.g. when trading currencies.

MarketRiskMeasure - To measure the volatility of the base value, you can either use the standard deviation or the Average True Range.

RiskTiming - Set the calculation period for the volatility measurement.

Reset - Here you can specify after which time you take your profits from the market and the invested capital is reset to the original initial capital. You can choose between **yearly**, **monthly**, **Never** (all capital is reinvested continuously) or **No Position Sizing**.

Instrument Count - Here you can enter the number of instruments in the chart. This way, a portfolio can be simulated in a single chart. It allows you to distribute your available funds across multiple instruments in one chart.

PERCENT VOLATILITY

This method works similar to the "Fixed Risk" method. It measures the volatility of the base value and scales the position sizing accordingly. As additional parameter, you have to enter a multiplier for the maximum expected loss. The result of the multiplication of this factor and the measured volatility is used as basis for the calculation of the position sizing.

OptionsFractionOfCapital - Enter how much fraction of capital is to be used for each trade.



Example for Percent Volatility

VolatilityModel - To measure the volatility of the base value, you can use the Average True Range or the momentum. The momentum is given as absolute value from one bar to the next one. With the result, a moving average over the "volatility period" is calculated for smoothing.

VolatilityPeriod - Enter the period over which the volatility is calculated.

BiggestVolaLoss - Enter a multiplier for the measured volatility. This value gives the highest maximum loss possible as multiple of the volatility.

Profit (Factor Profit) - With this multiplier, the amount of accumulated gain is included in the calculation of position sizes.

MaxLeverage - Enter the maximal leverage.

Reduce - Some of the options may lead to overstepping the originally calculated investment capital or risk parameters. For such cases, you can activate this option to ensure that the position sizes are reduced to meet the criteria of the main method.

Instrument Count - Here you can enter the number of instruments in the chart. This way, a portfolio can be simulated in a single chart. It allows you to distribute your available funds across multiple instruments in one chart.

MARKET'S MONEY

This method works similar to the Percent Volatility method. In addition to it, it reinvests profits with leverage. If your capital has decreased, the position size will be reduced.

If **FactorProfit** is set to the same value as **BiggestVolaLoss**, the method reacts identical to the "Percent Volatility" method.

Options



Example for Market's Money

FractionofCapital - Enter how much fraction of capital is to be used for each trade.

VolatilityModel - To measure the volatility of the base value, you can use the Average True Range or the momentum. The momentum is given as absolute value from one bar to the next one. With the result, a moving average over the "volatility period" is calculated for smoothing.

VolatilityPeriod - Enter the period over which the volatility is calculated.

BiggestVolaLoss - - Enter a multiplier for the measured volatility. This value gives the highest maximum loss possible as multiple of the volatility.

Profit (Factor Profit) - With this multiplier, the amount of accumulated gain is included in the calculation of position sizes.

MaxLeverage - Enter the maximal leverage.

Reduce - Some of the options may lead to overstepping the originally calculated investment capital or risk parameters. For such cases, you can activate this option to ensure that the position sizes are reduced to meet the criteria of the main method.

Instrument Count - Here you can enter the number of instruments in the chart. This way, a portfolio can be simulated in a single chart. It allows you to distribute your available funds across multiple instruments in one chart.

AUTOMATIC ORDER ROUTING

Under order routing, the routing of trading signals of a mechanical trading system to a connected broker is understood. The function can be used to:

- Route signals of a trading system automatically to the stock exchange via a broker
- Route signals of a trading system manually to the stock exchange via a broker
- Test a trading system under live conditions via the demo access of a broker

Please note that the automatic order routing is not available for every data provider.

Note:

Data vendors (and 3rd party data files) may deliver erroneous data that may be represented as valid data. Data vendors may change their interface at any time and without prior notice, such an update can introduce breaking changes. Access to data from a data vendor may be delayed or fail due to network congestion, network errors, server outages, power failure and other factors.

HOW THE ORDER ROUTING WORKS

To use the order routing function, a number of preferences have to be set:

- Create an account and configure settings in the Order Routing Options
- Activating the order routing for a chart or portfolio in the Automation Options of the Trading System
- Starting and stopping the order routing via the **Order Routing** toolbar, see the section Button Menu Order Routing.

Several levels of automation are possible, up to fully automated trading without interference from outside.

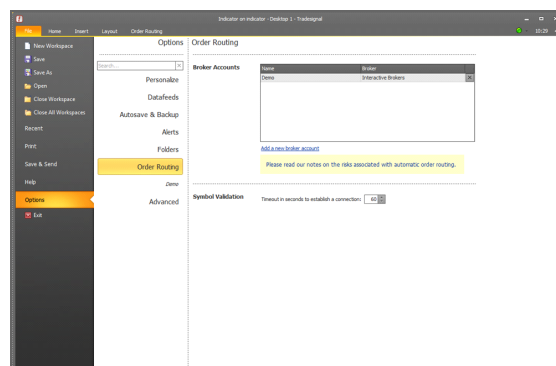
Important: When automatic order routing is active a small routing icon will be displayed above the toolbar, to the left of the clock. This icon may be clicked to **Stop All Order Routing**.

ORDER ROUTING OPTIONS

BROKER ACCOUNTS

A *Broker Account* is the terminology user in Tradesignal to refer to a connection to an account with a specific broker and associated settings. It is possible to have multiple named broker accounts to the same physical account at the same broker but with different settings. Doing so allows different strategies to be created for trading different security types with the same physical account for example.

Important: Some brokers do not provide a mechanism to take multiple manageable positions in the same security. Without this feature, it is not possible to *correctly* manage multiple *independent* strategy positions for the same security on the same physical account. Tradesignal will prevent order routing from being started if it is already active for a given security on the same physical account in



Order Routing in the Tradesignal Options

such cases.

- **Add a new broker account** - Accounts are added in the options menu under *File > Options > Order Routing > Broker Accounts* by clicking on the **Add new broker account** link.
- **Remove Account** - Accounts can be removed by clicking the **X** button next to the account in the list of accounts.
- **Edit Account** - Account settings can be modified by clicking on the account name listed below **Order Routing** under *File > Options*.

CONFIGURATION OF THE ORDER ROUTING ACCOUNT

In this menu, all settings for the broker account and the general handling of the orders are entered. Do not change the port number and IP address.

Selection of the Broker and name for the account are only available in the wizard when first creating the account.

- **Enter a name for the order routing account** - Enter an account name that is used for this broker within Tradesignal.
- **Port** - Port number of the broker's trading application, e.g. in IB TWS.
(Please note: If you use more than one TWS on one machine, you have to change the portnumbers in the TWS. After this Enter these Portnumber here. Otherwise the orders can appear in both TWS)
- **Localhost** - IP of the broker's trading application.
- **Account Code** - Your account at the broker, e.g. the "Interactive Brokers Account Code" for IB.
(Please note: The correct Account Number is only for synchronizing necessary. The Orders will be directed to the TWS via the correct Portnumber)*
- **Order Transmission** - Select here whether the orders are routed in the **Automatic** or **Manual** mode.
- **Action on Session End** - Configure here how orders shortly before the session end will be handled. They are either routed to the broker as a market order (**Map Close Orders to Market Order**) before session end (also see the section On-Close Order) or routed as **Map Close Orders to Market-On-Close Order**. In the latter case, the execution depends on the definition for this order type at your broker. A "Market Order" is routed instantly, while a "Market on Close Order" is routed at a certain time before the session end.
- **Log Level** - Here you can set the log level for details of the order processing and related tasks. The level **System** logs the least detailed data, the level **Details** the most. We recommend that you keep the default setting.

The information appears in the output window in the lower area of the window. It can be opened by a click on the grip bar.

Note:

Brokers may deliver erroneous data that may be represented as valid data. Brokers may change their interface at any time and without prior notice, such an update can introduce breaking changes. Instructions sent to a broker or access to quotes may be delayed or fail due to congestion, network errors, server outages, power failure and other factors. Broker-supplied software may be required for connection between Tradesignal and a broker. Such software may have stability issues or exhibit errors under some circumstances.

Order Routing Account Settings

ORDER CONFIRMATION

In this menu, you define whether Tradesignal should send the orders automatically to the broker or if certain constraints should be applied to the order routing.

Note that in the settings of the broker account, similar settings for order routing are entered. However, those determine the behavior of the order software of the broker.

Order Verification Settings in the Tradesignal Options

- **Manually confirm all orders** - Select this option for manual confirmation of automatically generated orders. There are several settings available, some of them depending on time. The options range from indefinite waiting to order cancellation. If this is not set, orders will be automatically sent.

The remaining settings configure exceptions to manual confirmation, namely:

- Send order cancellations without manual confirmation
- Send order modifications without manual confirmation
- Wait only a specific period of time to obtain manual confirmation and then take a default action:
 - Do not place order
 - Do not place order and stop order routing
 - Place the order

INITIAL SYNCHRONIZATION

In this menu, you can define which actions are to be undertaken during the start process of the order routing.

When starting the order routing, Tradesignal is able to synchronize the positions in the trading system with the account positions. Several options are available.

Order Start/Stop Settings in the Tradesignal Options

- Do not modify the account position - Positions are not changed.
- Modify account to strategy position with the first entry order from the strategy - The positions are only synchronized with the first entry order. This standard setting minimizes the trading costs.
- Cancel pending orders and modify account position with the first entry order - Pending orders are canceled and the positions are synchronized with the first entry order after the start. This may cause increased trading costs.
- Cancel pending orders and modify account position with a market order - Pending orders are canceled and the positions are instantly synchronized with an order. This may cause increased trading costs.

STOPPING ORDER ROUTING

Here you can set what is to happen when the order routing is stopped.

- Cancel pending orders - Pending orders are canceled without closing positions.
- Cancel pending orders and close positions - Pending orders are canceled and open positions are closed.
- Do not modify account position - Positions are not changed.

SYNCHRONIZATION

Here you can define how Tradesignal will react when the account positions are no longer synchronized with the positions in the trading system. Several settings are available, from automatic synchronization with market orders to the stop of the automatic order routing with an alert.

If the account and strategy positions are no longer synchronized:

- Cancel all pending orders and generate a market order to synchronize positions - Pending orders are canceled and a new order is issued to instantly synchronize the positions. This may cause increased trading costs, especially if the synchronization is frequently lost.
- Cancel all pending orders, generate exit orders and synchronize with the next entry order - Pending orders are canceled, current positions are closed and new entry positions are opened. This may cause increased trading costs, especially if the synchronization is frequently lost.
- Synchronize position on the next entry order - Pending orders remain open, and the positions are synchronized only with the next entry order issued from the trading system. The entry order is changed (increased/decreased) to include the position synchronization. Note that this can only be done with a market order or one limit/stop order. Systems with several limit orders, e.g. with limit order above and below the current quote, cannot be synchronized this way.
- Cancel all pending orders, close account position and synchronize position with the next entry order - Pending orders are canceled, current positions are closed and new entry positions are opened. This may cause increased trading costs, especially if the synchronization is frequently lost.
- Stop automatic order routing and notify user with an alert - The order routing is stopped (any pending orders and positions remain open).
- Notify user with an alert - The order routing is not stopped but an alert is issued.
- Do nothing
- Seconds to wait for account and strategy positions to synchronize - Enter the duration in seconds that the synchronization may take. If synchronization fails, the timeout counter below starts. An out-of-sync chart will be highlighted.
- Seconds to wait for account and strategy positions before synchronization timeout - Enter the duration in seconds before the above entered settings for handling the synchronization loss are applied.

Note: If the **Send On-Close orders a few minutes before session end** option is enabled, the position sizes will not be checked within the specified number of minutes before session end. As soon as the session is closed, Tradesignal will continue to check the position sizes in order to ensure the strategy and account positions are synchronous.

STOP AND LIMIT ORDER

Tradesignal offers you two methods for how generated stop or limit orders are to be treated at the session end. These options determine how much the back test results may differ from real trading results.

Position Synchronization Settings in the Tradesignal Options

- Wait for Tradesignal to report a fill then send as a Market Order - Tradesignal issues a stop or limit order only after the last order was confirmed as filled by the broker.
- Send the order to the broker as-is; take no actions if the order is not filled as expected - Tradesignal issues stop or limit orders independently of fill confirmations.
- When using the IB TWS, this option should be deactivated. When the option is deactivated, the existing stop or limit orders are updated at every new bar/tick in Tradesignal. If the option is active, all existing orders are canceled and new orders are issued. This may result in problems, especially for fast-trading stocks or futures, as the orders cannot be canceled quickly enough by the trading software.

ON-CLOSE ORDERS

Here you can define how an order is routed when it is classified as "on-close" by the trading system.

- Send on-close orders at the end of the session
- Send On-Close orders a few minutes before session end - Select this option if the on-close order is to be issued before the set session end. You can enter a time in **Number in Minutes** or click on the spin buttons. Please note that you can set similar order routing settings in the account settings of the broker. If your broker supports real on-close orders, you may be able to send them right at the session end.
- Cancel active limit and stop orders after on-Close orders are generated - Select this option if unfilled stop or limit orders are to be canceled when an on-close order is generated. This will keep the system from accidentally filling several orders.

TIME-IN-FORCE

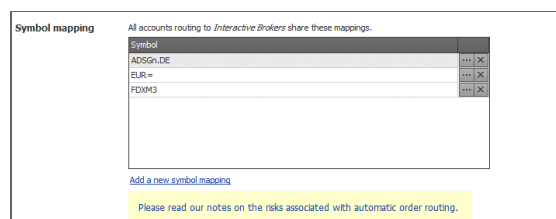
- Orders should stay active until canceled - If selected, the order may stay active for longer than the trading day.
- Orders should stay active for the trading day - If selected, the order is only active for the current day and deleted at session end, if not filled.
- Always generate new orders when a bar opens - Select so that pending orders are canceled and new orders are generated at each new bar. This option should not be used for short time periods, but may make sense for trading systems with long periods, e.g. weekly.

SYMBOL MAPPING

"Symbol Mapping" in Tradesignal means mapping the symbols of the online data provider and the broker that refer to the same underlying instrument.

For the order routing to work, the program has to know this mapping. The correct settings are very important for trading.

Attention: If the wrong symbols are entered or mapped here, the wrong symbol may be traded by the broker in the worst case.



Symbol	Data Provider	Broker
ADSGn.DE		X
EUR =		X
FDXM3		X

[Add a new symbol mapping](#)

Please read our notes on the risks associated with automatic order routing.

Symbol Mapping

EDITING SYMBOL MAPS

- **New** - Enter a new symbol map. A dialog opens, see next section.
- **Edit** - Edit the selected symbol map.
- **Delete** - Delete the selected symbol map.
- **Delete All** - Delete all symbol maps.

CONFIGURING ORDER ROUTING SYMBOL MAPPING

In this dialog you can create a new symbol map or edit an existing map.

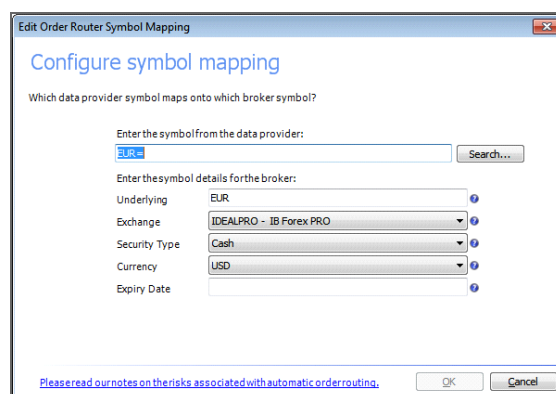
Enter the symbol from the data provider:

- When a chart with a symbol is open and selected, this symbol is automatically entered here. Otherwise, click on the **Browse** button to start the symbol search in Tradesignal. Note that many symbol names are a combination of symbol shortcut and exchange name.

Enter the symbol details for the broker:

- **Underlying** - Enter the asset's underlying ticker symbol.
- **Exchange** - Select the exchange from the list at which the symbol is traded.
- **Security Type** - Select the symbol type from the list. Most brokers use this setting to assign the shortcut to the right exchange.
- **Currency** - Select the currency delivered against the purchase of the underlying currency.
- **Expiry Date** - Enter an expiry date for the underlying, if available.

Attention: Please check the settings to ensure that the right symbols are ordered.

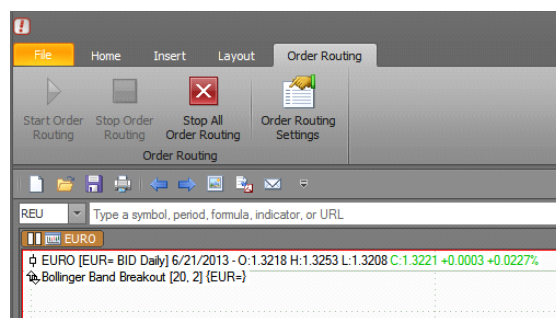


Order Routing Symbol Mapping Dialog

ORDER ROUTING TOOLBAR

Open this toolbar by clicking on the **Order Routing** tab above the toolbar.

- **Start Automatic Order Routing** - Start the order routing for the selected chart or portfolio. To start all order routings, press the **Ctrl** key while clicking.
- **Stop Automatic Order Routing** - Stop the order routing for the selected chart or portfolio.
- **Stop All Automatic Order Routing** - Stop the order routing for all charts and portfolios.
- **Order Routing Options** - Opens the Order Routing Options in the Tradesignal Options.



Button Menu Order Routing

STRATEGY AUTOMATION OPTIONS

This menu is opened by clicking on the small plug symbol on the lower right of the chart or a portfolio. Here you can activate different levels of automation for the order routing. The following settings are available:

- **No Order Routing:** Select this option if no automatic order routing is used. No signals are output by the strategy. The Position Manager remains empty.
- **Position Manager Only:** Select this option to send the signals generated by the strategy to the Position Manager. To submit the orders, you have to enter them manually into your broker's order software.
- **Broker:** Select this option to route the orders automatically. The signals of the trading system are displayed in the Position Manager and also routed to the broker. The limitations for this automation are set in the Order Verification settings of the Order Routing Settings in the Tradesignal Options, see above.



Automatic Order Routing Menu

After selecting an option, you can start the order routing from the **Order Routing** button menu. Click on the **Start Automatic Order Routing** entry.

CONTROLLING HOW AND WHEN ORDERS GET FILLED

When using order routing the one of the major issues is ensuring that fills generated in the strategy closely match fills in the brokerage account. The Money Management property **Realtime Price Source** can be used to tell Tradesignal to use best bid and best ask prices from the Broker instead of using last prices from the data source. The following settings are available:

- **Bid/Ask** - Tradesignal uses the best bid / best ask data provided by either the broker (if an order routing module has been connected) or the data provider. If neither the broker nor the data provider is delivering bid/ask prices, Tradesignal uses the last price instead (for example in the case of indices). Best bid and best ask prices are generated when a market participant wants to buy or sell a number of shares/contracts while a last price represents an actual trade that has been made. Therefore this option has the benefit of more accurate fills for your orders. The caveat is however that the results collected in real time will not match the results of back-testing which will always only use the last price.
- **Last** - Tradesignal uses the last price delivered by the data provider. This is the same price as the one used to back-test the strategy. A last tick represents a trade that has been made in the market. When using this mode, Tradesignal can only fill and order after a trade has been made, even if there was a matching offer/request in the market prior to the trade. This mode ensures that real time results are closer to back-test results.

⋮ MISCELLANEOUS FEATURES

TRADESIGNAL OPTIONS

In the Tradesignal options, you can find general settings for Tradesignal, for example

- preferred view and chart type
- data connections and order routing
- timescale and scope of data backup

You can open the window *Tradesignal Options*

- by clicking on the **File** menu in the header bar and selecting **Options** from the menu, or
- by double-clicking on the connection icon above the toolbar.

In the following chapter, you will find information about the settings or links to other chapters with detailed information.

SEARCHING THE OPTIONS

In addition to drilling down search for an option by label, it is possible to use the search box located to the top left of the options pane to search for an option by keyword

1. Enter the search term, e.g. "Excel"

The available options matching the keywords will be displayed as you type

To reset the search click the **X** button next to search box, or click on a options category.

PERSONALIZE

ON STARTUP

Which workspaces to open automatically on start up

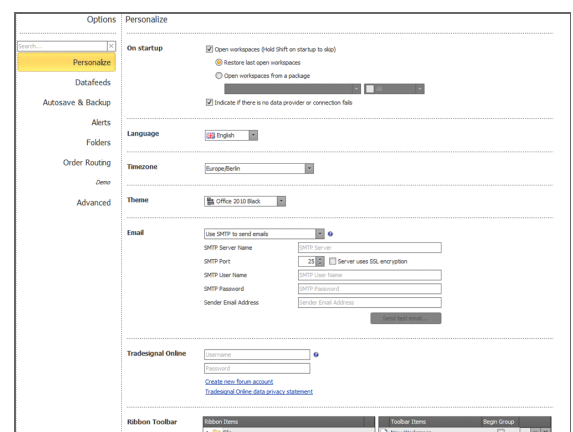
LANGUAGE

Which application language to use, changing this setting will require a restart of the application.

TIMEZONE

Set the local timezone. This setting is used to adjust time values in charts to the local timezone.

THEME



Tradesignal options - start view

Which application theme to use.

EMAIL

The settings to use when the application needs to send an email. Two methods exist

- **SMTP** - Requires the SMTP servers details to be added. Most reliable solution for sending emails from the application.
- **MAPI** - Will use the locally installed email software to send emails. May cause problems with security settings.

TRADESIGNAL ONLINE

Log-in details for the Tradesignal Online website. The settings in this area are necessary if you want to publish material (such as a chart) to the forums on Tradesignal Online (see the chapter Publishing).

- **Username** - Enter your Tradesignal Online user name.
- **Password** - Enter your Tradesignal Online password. For security reasons, this is not displayed.

Click on "Create new forum account" to open the website account.

RIBBON TOOLBAR

Tool to configure the Quick Access Toolbar. All possible functions are listed in a tree in the left pane. These functions can be added to the right pane and will appear in the Quick Access Toolbar above the regular toolbar. Options next to each Toolbar item allow it to be repositioned, removed or grouped.

PRINTING

Options to configure standard printing options, most notably the default watermark and header/footer layout.

RESTORE DEFAULTS

Clicking on this button will restore all system settings to their default values, a dialog will be displayed to confirm this action.

Individual settings may be restored to their factory setting by right-clicking on the label for the option and selecting **Reset**.

DATAFEEDS

Detailed information about connecting online and offline data can be found in the chapter Data Sources.

AUTOSAVE & BACKUP

AUTO SAVE

- **Automatically make a periodic recoverable backup of any open documents** - This is the default setting. When selected, the work is saved in regular intervals (given by the parameter **Frequency in minutes to save open items**) and available as backup. You can edit the frequency value.

APPLICATION BACKUP

Here you can enter settings for the backup of your current configuration. The following information is saved:

- Packages with all their files, including information whether they are write-protected (date and time are not saved!)
- Symbol lists
- Symbol settings for automatic order routing
- User settings
- Edited quotation data

This way you can save the current state of Tradesignal and then use the resulting backup to restore your settings on another computer - or on your computer, for example after a new installation of the operating system.

The following options are available:

- **Backup Location** - This is the folder where the backups are saved to. To enter another one, click on **Browse** and select one in the Windows file dialog.
- **Number of recent backups to keep** - This is the maximum number of backups that will be kept. When going beyond this number, old backups will be overwritten, starting with the oldest. Max. number is 100.
- **Create a backup when the application shuts down** - When selected, the data is saved as backup before shutdown.
- **Create daily backup at** - Here you can enter a time at which the backup will take place. Note that this only will be done as long as the application is running.

Click on **Create Backup Now** to create a backup for the current settings and data.

Click on **Restore Backup** to restore an old backup. Select one of the available backup files from the dialog. Then choose whether you want to:

- **Restore all files including settings** – restore all saved data as listed above.
- **Restore all package files** – restore the package files only.

In addition, you can select the option **Restore exclusively missing files** so that already available files are not overwritten. Confirm the warning. After the restoration, the application is closed so that the new settings can take effect upon restart.

Note that all open workspaces have to be closed before restoring a backup. This is done fastest when you select **Close All Workspaces** from the file menu.

ALERTS

Detailed information about the alert setting options can be found in the chapter Alerts.

FOLDERS

PACKAGES

Detailed information about the package options can be found in the chapter Package and File Management.

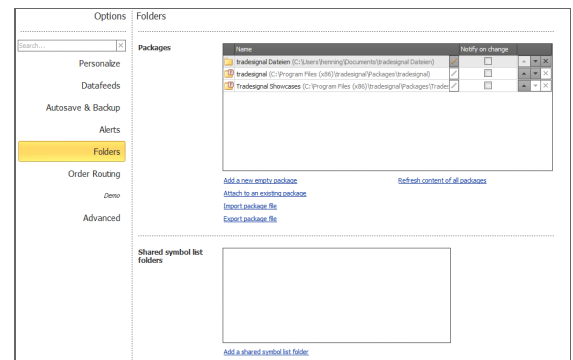
SHARED SYMBOL LIST FOLDERS

On this page you can add/remove folders containing symbol lists in XML format as exported from Tradesignal.

By having these folders on a network share, you can share symbol lists between users.

EQUILLA PATHS

- **C-API Extension DLLs** - In this folder, special Equilla interface files are saved. Do not change this path.
- **PrintToFile output** - In this folder, files are written by Equilla scripts (see the chapter Data Output via File Interface).



Shared Symbol Lists Folders

RSS FEED

Here you can configure RSS feeds. RSS feeds are added to the news displayed in the news window.

- **New** - Add a new feed. A dialog opens where you can enter the URL of the feed. Make sure that it starts with "http://".
- **Edit** - Edit the URL of the selected RSS feed.
- **Delete** - Remove the selected RSS feed from the list.

ORDER ROUTING MODULES

Folder where order routing modules are located.

ORDER ROUTING

Detailed information about the order routing options can be found in the chapter Order Routing Options.

ADVANCED

Many options to configure default behavior are found in this section organized by the following categories

EQUILLA EXTENSION

These options are available in and offer the option to enable or disable support for Equilla extension DLLs.

- **No extensions allowed**
- **Only C API extensions allowed** - Allows Equilla extensions that use the C API. This is the default setting.
- **Only COM API extensions allowed** - Allows Equilla extensions that use the COM API.
- **Both C and COM API extensions allowed** - Extensions for both APIs are allowed.

CHART STYLE

- **Default style** - The chart type that will be used for all newly created charts (excluding tick-by-tick charts).
- **Tick chart style** - The chart style to use for new tick-by-tick charts.
- **Value scale mode** - The value scale mode to use for newly created charts (Linear or Logarithmic).

INSTRUMENTS

This section allows configuration of the default history length to use for charts and other document types in various periods.

- **Default period** - The default period to use for newly opened charts and lists. The default period is daily.
- **Open instruments in a new subchart** - Check this box to always add new instruments as a sub chart to an existing chart. Uncheck the box to always add the new instrument to the main chart.
- **Resolve ambiguous symbols** - When a symbol is entered on the command line or in an inline instrument, this option determines how the symbol should be resolved to an instrument if the symbol is multiply defined.

WORKSPACES

- **Maximum number of open workspaces** - Increase this number to allow more simultaneously open workspaces, the default is 30. Please note that opening more workspaces will require more system resources.
- **Maximum number of items in a workspace** - increase this number to open more items in a workspace. The default value is 30. Please note that opening more items in a workspace will require more system resources.
- **Show close button on selected workspace tab**
- **Hide titles of non-selected workspace items**
- **Show unremove bar at the bottom of the screen** - When not checked the unremove option will be displayed at right of the application's title bar.
- **Automatically close a floating desktop when the last workspace has been closed**
- **Set the workspace tab button size**

PROPERTY INSPECTOR

- **Always show properties when a new item is selected**

COMMAND LINE

- **Generate OHLC outputs for one-line formula-based symbols** - When this item is selected and a spread formula is entered on the command line, an instrument will be created with open, high, low and close values based on the entered

formula, instead of just using the close value.

- **Display security name for one-line formula-based symbols** - When this option is checked, the application will substitute a symbols display name for the symbol in formulas.
- **Correct symbols used in formulas used on the command line** - This option prepends the prefix to a symbol used in a formula if it has been forgotten.
- **Clear command line history**

SYMBOL LISTS

- **Show Symbol Lists toolbox after inserting a chart into a workspace**
- **Use preferred period and FID when opening new charts from a symbol list** - Some symbols in a symbol list may also specify a preferred period and FID. If they do, and this option is selected, the instrument added to a chart will override the charts default FID and period and use the preferred settings instead.
- **Show all symbols collected on a DataConnect server in a separate symbol list.**
- **Symbol display format** - How symbols in the Symbol List toolbox show be displayed.

PERFORMANCE

- **Prevent system suspension/hibernation while the application is running**
- **Reduce the evaluation frequency for non-visible charts**
- **Maximum visible instruments** - The maximum number of instruments that may be simultaneously open.
- **Minimum memory for optimizer/scanner (MB)** - The minimum amount of system memory that must be available to start a new scan or optimization task.

INDICATORS & STRATEGIES

- **Automatically add the strategy equity indicator when strategies are added to a chart**
- **Do not show the input mapping dialog if all inputs can be automatically mapped**
- **Add color-tagged indicators to the ribbon** - If an indicator in the Indicator toolbox is color-tagged using the context menu (yellow, blue or red) and this option is checked. A button will be added to the *Favorites* group of the ribbon that when clicked will add the indicator to a chart.
- **Add color-tagged strategies to the ribbon** - If a strategy in the Strategies toolbox is color-tagged using the context menu (yellow, blue or red) and this option is checked. A button will be added to the *Favorites* group of the ribbon that when clicked will add the strategy to a chart.

Clicking on the **Recompile all user indicators and strategies** will force all user indicators and strategies to be recompiled. Progress will be shown and the result of the compilation indicated.

SESSIONS

It is possible to set here exactly when On-close orders are generated in order routing.

CSV FILE EXPORT

- **Field separator when exporting data** - Default is a semi-colon character.
- **Language used for formatting numeric values**
- **Enclose values in quotes**
- **Export all optimizer runs in the optimizer results**

CLIPBOARD

- **Field separator when copying data** - default is a tab character.
- **Language used for formatting numeric values**
- **Enclose values in quotes**

TEMPLATES

- **Allow to update the templated charts when the base template change** - If this option is checked, charts that have been created based on a template will automatically be updated when the templated is changed and saved.

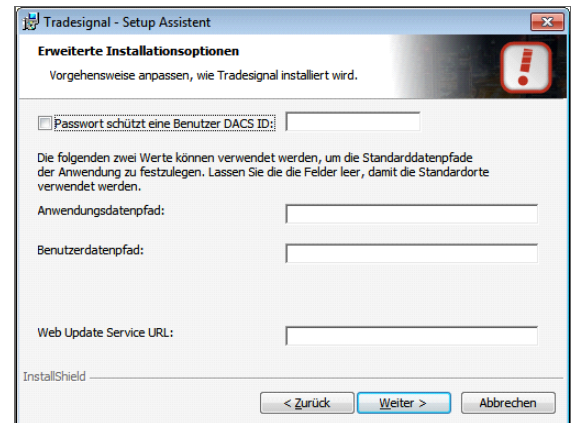
ADVANCED INSTALLATION OPTIONS

The information in this article is only relevant for system administrators who need to use different settings or additional features of Tradesignal.

During installation of Tradesignal you can select **Show the advanced installation options page (recommended only for administrators)** which allows you to specify some additional options.

PASSWORD PROTECT DACS ID

With this option you can select a password for the DACS ID.



Advanced Options during Installation

DATA PATHS

You can change both the application data and the user data path.

If you leave either of these fields blank, the default settings will be used.

SHARED DATACONNECT SETTINGS

You can specify the file location of an INI file containing connection settings for the Tradesignal DataConnect and external data sources. This way there is a central point at which to configure connection settings for all your installations and new installations are easily configured to use your existing data sources.

This option sets the registry key *HKLM\Software\SystemSoft\TradeSignalEnterprise\ConnectionSettingsFile*.

UPDATE WEB SERVICE

This allows you to specify an URL which is to be checked periodically for updates of Tradesignal.

If you select this option, the registry key *HKLM\Software\SystemSoft\TradeSignalEnterprise\SiteUpdateService* will be set.

ALERTS

A very important feature of charting software is the alert function. It helps you keep an overview even when working with a large number of running processes and functions. Use the alert function in Tradesignal for the following:

- Trading systems that alert you of certain activities
- Indicators that offer an inbuilt alert function in their codes
- The chart tools trend line, trend channel, regression channel, stop line and the Fibonacci Retracement
- Your own programs in which you can include offer alerts

ALERT WINDOW

There is a specific alert window on the bottom of the workspace. The displays differ slightly depending on the alert category. In Tradesignal you can choose which columns to display for each category by selecting *Add/Remove columns* from the alert window's context menu.



Alert area in the workspace

- Alerts coming from chart tools or indicators are listed in the "Indicators & Tools" tab, with the triggering symbol and reason, for example a trend line crossing or an indicator crossing a signal line.
- Alerts coming from trading systems are listed in the "Strategies" tab. Here, the triggering symbol and all events of the trading system are listed, like opened or canceled orders, and also feedback from your broker in case of activated automatic order routing. The latter depends on your settings in the Order Routing options.
- Application level alerts such as losing the connection to your data feed are shown in the "Application" tab.

Each tab will show the number of unread alerts next to the tab's icon.

MAXIMIZING THE ALERT PANE

The small maximize icon shown directly below the alert pane tabs can be used to maximize the alert pane to fill the entire main application window (with the exception of Toolbar and toolboxes). Clicking the same button a second time will restore the alert pane to its original position.

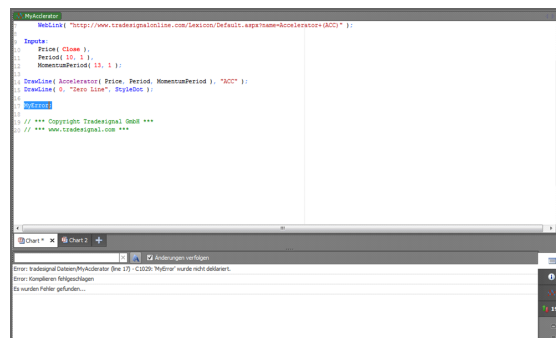
COLLAPSING THE ALERT PANE

The small minimize icon shown below the alert pane maximize button can be used to collapse the alert pane. When collapsed the alert pane will be shown as a single button (bearing an *i* symbol) to the far right of the workspace tabs. If there are any unread alerts, they will be displayed next to this button.

OUTPUT WINDOW

The tab labeled "Output" contains the output of Equilla's **Print()** command and also the errors and warnings generated by the Equilla compiler. The text box at the top allows filtering of its entries to show only matching lines.

The output window contains a maximum number of 100,000 entries. If you need a larger number of entries, you can edit the file tse.ini in the application data folder of Tradesignal by adding a value up to 10,000,000 for the setting **MaxOutputMessage** in the **General** section. However be advised that increasing this number may negatively affect your performance.



Output area in the workspace showing compiler errors

ALERTS IN TRADING SYSTEMS

ALERT FUNCTIONS IN INDICATORS

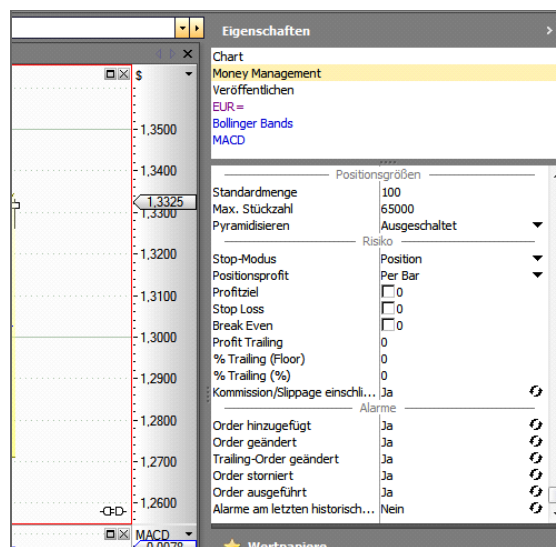
Many indicators included in the software are equipped with an alert functionality. As default, the alerts are deactivated. However, you can switch on the alerts by opening the property inspector in the toolbox and setting the option **ShowAlerts** to **Yes**. As of that moment, signals like line crossing or extremes are reported as alerts.

ALERT FUNCTIONS IN MONEY MANAGEMENT

The Money Management alerts in four different situations:

- **Order added**
- **Order changed**
- **Order cancelled**
- **Order filled**

The alerts are reported by the trading system.



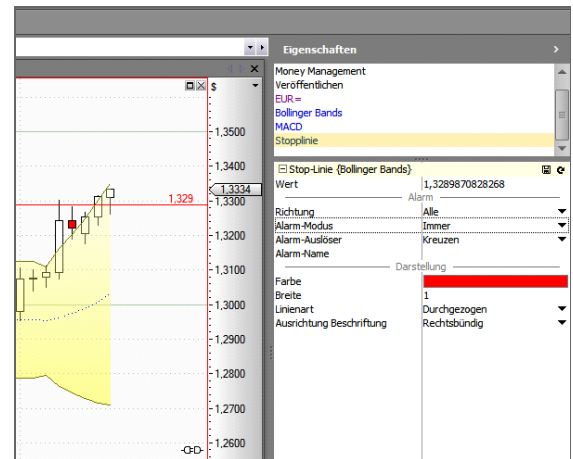
Detail: Alerts in the money management in the toolbox

ALERT FUNCTIONS IN CHART TOOLS ALARM

SET AN ALERT MODE

For the chart tools trend line, trend channel, regression channel, stop line and Fibonacci Retracement, the following alert modes are available:

- **Always** - An alert is always set off when the condition is met, e.g. a line crossing. In realtime operation, this can result in multiple alerts from the same tool, if the stock crosses the line several times.
- **Once Per Bar** - An alert is only set off once per Bar. For example, if the price crosses the threshold several times during the period, only one alert is issued and all others are ignored. Once a new Bar has been created, a new alert will be issued when the condition is met.
- **Once** - An alert is set off only the first time the condition is met. All other events are ignored.



Choice of Alert Mode in the Chart Tools

SET AN ALERT DIRECTION

You can also set an alert direction for the chart tools to define in which "direction" the crossing has to take place to set off an alert.

- **All** - Alerts are set off no matter whether the lines are crossed downwards or upwards.
- **Support (downwards)** - Alerts are only set off in the case of a downwards crossing.
- **Resistance (upwards)** - Alerts are only set off in the case of an upwards crossing.

GLOBAL ALERT SETTINGS IN TRADESIGNAL

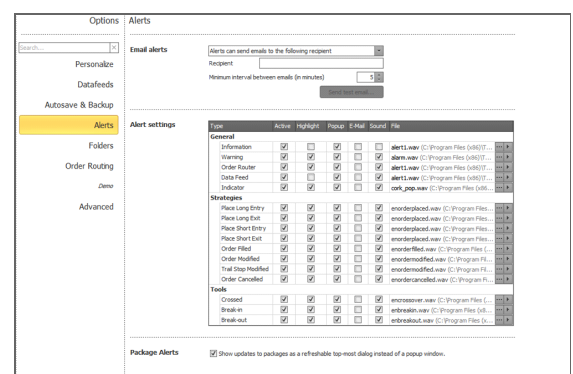
In the Tradesignal Options, you can define several global settings for the alert output, like an E-Mail address or the behavior for each type of alert.

For this, open the file menu and select the **Options** entry. Then choose the area **Alerts** in the Options window. The following options are available:

Alert options which can be configured per alert type:

- **Active** - Here you can enable or disable alerts of this type.
- **Highlight** - With this option enabled, alerts are highlighted in the alert window and of **Popup** is enabled too, the popup window will not close automatically.
- **Popup** - If this option is enabled, the alert is displayed in a special popup on the bottom of your screen. If **Highlight** is also enabled, the popup will be shown until closed, otherwise it will hide after a short amount of time.
- **E-Mail** - Enable this to send an E-Mail when an alert of this type is triggered. It is only sent when **Send an E-Mail when an alert occurs** is enabled.
- **Sound** - Here you can select whether an alert sound shall be played for each alert. Click on ... to choose a different sound file from a file dialog. To preview the sound, click on the triangle button.

General settings:



Alert Settings in the Tradesignal Options

- **Send an E-Mail when an alert occurs** - By unchecking this, you can prevent E-Mails from being sent when an alert is issued regardless of the setting per type. Select whether you want to send via the MAPI or SMTP protocol and enter the SMTP configuration, if necessary. You can send a test E-Mail and also set an interval (shortest time between two sent E-Mails).

PACKAGE AND FILE MANAGEMENT

Packages correspond to folders on the computer and offer some additional functions for organizing and managing your data.

You can use packages for the following:

- Setting up a folder structure in the Tradesignal data folder
- Sharing data with other users
- Saving data for backup
- Export Data

The following data can be saved in a package:

- Indicators
- Strategies
- Functions
- Templates
- Saved workspaces

Packages are compressed on export to reduce the file size. Equilla functions are automatically added to the package if necessary.

When working with packages, please remember that they correspond to physical folders on your computer. Therefore, you can - by intention or by accident - rename, move or delete these folders with a program like the Windows Explorer. The result of such actions may be that the packages are no longer visible in the package management of Tradesignal. As long as the folders still exist, however, you can set up packages again, see the section Package Management in the Tradesignal Options.

NEW PACKAGE

- In the toolbox, click on **New Package** in the area *Related Tasks* to set up a new, empty package. A file dialog appears in which you can either choose an existing folder or set up a new one via the button **New Folder**.
- Alternatively, you can set up a new package by way of its contents, as shown in the following example.

EXAMPLE: SETTING UP NEW INDICATORS IN A NEW PACKAGE

1. Click the button **Indicators** in the toolbox.
2. Click on **New Indicator** in the *Related Tasks* area. The Create Item Wizard opens.
3. Choose the option **Download an existing indicator from Tradesignal Online**.
4. Select one or more indicators.
5. Confirm with the button **Done**.
6. You can then either save the new indicators in an existing package or in a new one by clicking the button **New Package**. Start a new package.

Tradesignal sets up a new folder in which it saves the downloaded indicators. The new folder can be managed with the package and folder management in Tradesignal and also with the normal folder functions of the Windows file system.

SHOW AND HIDE PACKAGES

With the help of packages you cannot only set up a file system with a subfolder structure, but also organize the contents of the toolbox windows. For the latter, however, the packages have to be displayed.

You can toggle the package display by clicking on **Show/Hide Packages** in the *Related Tasks* area.

- If packages are shown, the objects (indicators, strategies, workspaces or Equilla functions) are sorted by packages. This is of little importance as long as you are only using the packages provided by Tradesignal.
- If packages are hidden, all objects are listed alphabetically without any further breakdown.

MOVE PACKAGE CONTENTS

You can move contents like indicators or functions between packages. For this, the packages have to be displayed.

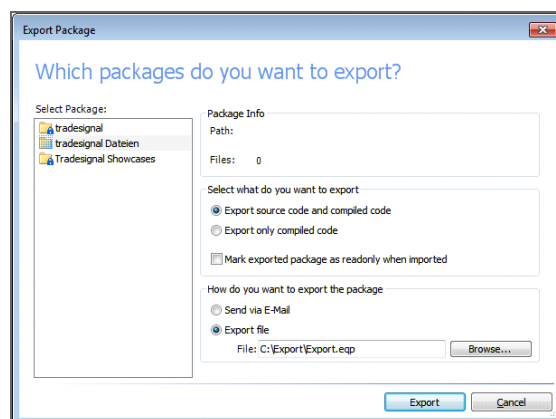
1. Open a tool window in the toolbox, for example **Indicators**.
2. Select an entry in the list and drag and drop it from one folder to the other.

The entry (e.g. the indicator) is moved between the Tradesignal packages as well as physically between the folders on the computer.

EXPORT OR MAIL PACKAGES

Packages can be exported. During export, the folder with all its contents (workspaces, indicators etc.) and all necessary Equilla functions is saved as a compressed file. You can then use this file as a local backup or mail it to other users for sharing.

1. Select *File > Options > Folders > Packages > Export Package file*. A dialog opens.
2. In the *Choose Package* area to the left, select the folder to be exported. The list will show all folders that are registered by Tradesignal. (It is possible that folders below the main folder *Tradesignal Files* will not be shown because they were hidden in the package management, see below.)



Export a package

SET THE EXPORT OPTION

Select what you want to have exported:

- **Export source code and compiled code** - Select this option if the source code should be included, e.g. when sharing the package content with another user.
- **Export only compiled code** - Select this option if the source code should not be included upon export, e.g. if you want to

share the data with other users without giving them access to the original source code.

- **Make exported packages read-only when imported** - If you select this option, the contents of the package cannot be edited after import.

SET THE EXPORT ROUTE

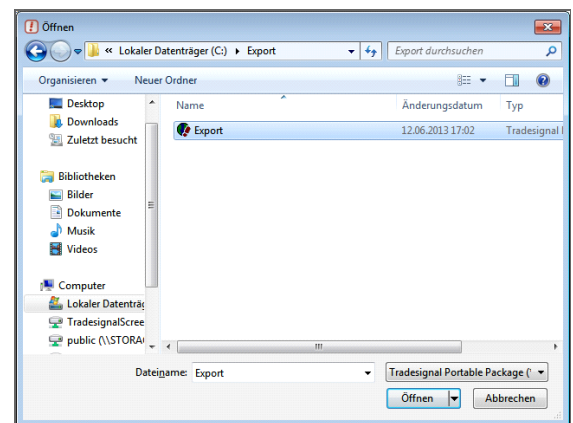
- **Send via E-Mail** - If you select this and then click the **Export** button, your default mail program will be opened and the file will be attached to a new mail.
- **Export file** - Select this if you want to save the file to your computer. ""
- **Browse** and selecting another one (or changing the file name). Click the **Export** button to start the export.

IMPORT PACKAGE

An exported package can be imported into Tradesignal. For this, the package file has to be available on the hard drive of your computer or on a removable storage drive that can be accessed from your computer.

You can import a package by double-clicking on it in Windows Explorer. (If you received the package as an E-Mail attachment, you can either save it on your desktop or import it directly into Tradesignal by double-clicking on it.)

Alternatively, you can select *File > Options > Folders > Packages > Import package file*. Select the package file in the file dialog and confirm the import.



Import a package

The program then decompresses the data in the package file and creates a new folder with the package name below the main folder *Tradesignal Files*. The new folder will appear in the toolbox window and the package management.

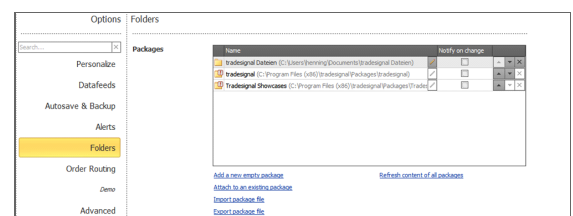
PACKAGE MANAGEMENT IN THE TRADESIGNAL OPTIONS

Tradesignal offers you several functions for the management of packages.

To access these functions, navigate to the *File > Options > Folders > Packages* area.

You have the following options:

- **Add a new package**
- **Attach to an existing package** - Access a previously hidden package. In this case, a file dialog opens in which you can find and select the package.
- **Remove** - If you select a package and click the **X** next to it, you can either choose



Package management in the Tradesignal Options

- **Remove this package, but leave its contents untouched so that other users can still access them.** In this case the package is only removed from the folder structure in Tradesignal but the physical folder is still available.
 - **Remove this package and permanently delete its contents.** In this case, all data will be removed from Tradesignal and also physically deleted from the hard drive.
- **Rename** - Select the package and click the edit (pen) icon. The function renames both the package and the folder on the hard drive.
- **Move up/Move down** - With these buttons you can change the folder sequence in Tradesignal. Note that if you have functions/indicators/strategies of the same name in different folders, changes in sequence may have effects on the processing order. When compiling an indicator/strategy, Tradesignal tries to find the necessary functions/indicators in the same package. If they cannot be found there, all packages are searched according to the given sequence, and the first find is used for the compilation.
- **Alert on Update** - Check the box if you want to be alerted to updates to the package. In combination with the checkbox *File > Options > Alerts > Package Alerts > Show updates to packages as a top-most dialog with refresh option* this allows you to refresh packages as soon as they are updated.
- **Refresh contents of all packages** - The contents of the package folders are checked by Tradesignal only upon start. Therefore, if Tradesignal is running and files are changed outside of Tradesignal (added, renamed, or deleted), you need to click **Refresh** so that the package contents are updated in Tradesignal. This is especially important in cases such as:
 - While Tradesignal was running, you received a workspace attached to an E-mail and saved it into a package folder.
 - You are working with packages on a network drive and other users changed the package contents.
- You can find the **Refresh** function also in the context menu of packages in the toolbox.

As a partial alternative to package management in the Tradesignal options, you can also create, rename or delete packages in the *Open* dialog. You can call up this dialog by opening the file menu and selecting **Open**, or by pressing the key combination **Ctrl + O**.

PRINT

Your charts, strategies, indicators or Equilla codes do not have to be limited to your computer. All documents can be published right out of Tradesignal (as printouts or files), for example in Internet forums, print media or by E-Mail. In the other direction, it is also possible to import documents in Tradesignal from Internet forums or by E-Mail .

This way, you can exchange content with friends, colleagues and supervisors and work on Equilla code together, for example.

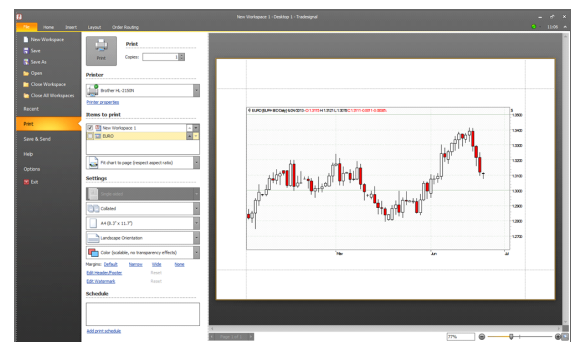
For exchanging complete packages, please see the chapters Export Packages and Import Packages.

PRINTING CHARTS AND TABLES

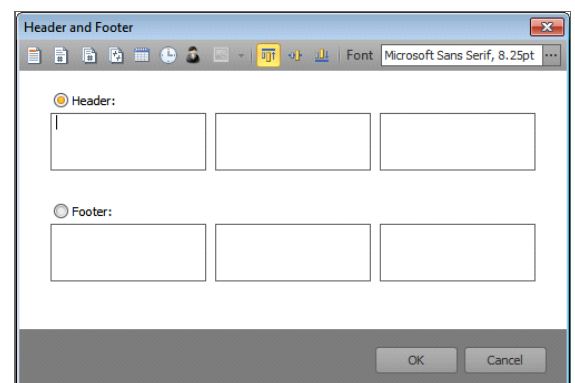
You can print all documents (Charts, Scanner, Watchlist, Equilla code etc.). Click on the **Print** button in the *File* menu. The Print Wizard opens. Here you can choose what in the currently selected workspace should be printed.

After selection, a *Print Preview* pane opens. Here you have the following options:

- **Print** - Start the printing. The standard Windows printer dialog opens.
- **Copies** - Number of copies of the workspace to print.
- **Printer** - The printer to which to print.
- **Printer Properties** - Advanced configuration of the selected printer.
- **Items to Print** - List of items in the workspace to print (including an extra page containing the workspace as a whole. The items are printed in the order listed. The arrow buttons can be used to change the print order of the items.
- **Chart Scaling** - Determines how charts should be scaled to fit the printed page.



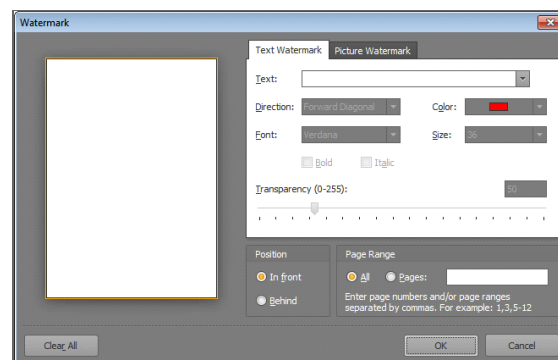
Print Overview



Edit Header/Footer

- **Duplex Mode** - Depending on the duplex/simplex capabilities of the printer.
- **Collation** - Options to control how pages are collated.
- **Paper Size** - Selection of the paper size.
- **Orientation** - Portrait or landscape.

- **Color Mode** - Select whether the document should be printed in **Full Color (with transparency)** (24 bit RGB with alpha channel), **Full Color (without transparency)** (24 bit RGB without alpha channel) or as **Black and White**. If the image is scaled on your printer (e.g. into a larger format), select **Full Color (without transparency)** for better results.



Edit Watermark

- **Margins** - Some predefined margin options. Margins can also be manually adjusted using click and drag directly in the preview pane.
- **Edit Header/Footer** - Shows the header/footer dialog where text and various macros can be set for the page header and footer. The alignment of text and font can also be changed. The following macros are available:
 - Document title
 - Page Number
 - Page Count
 - Print Date
 - Print Time
 - Print User
- **Edit Watermark** - Show the watermark dialog where a watermark based on text or an image can be specified for each page on the printed output. The watermark dialog presents many options to configure how the watermark is displayed, including
 - Text, Text direction, Font, Color and Size.
 - Image file and alignment.
 - Translucency.
 - Positioned in front or behind.
 - Page range to apply watermark to.

For a chart and performance report, you can find additional print-out options in the toolbox, category **Publish**. You can change the axes' colors and the header.

The print options selected for a workspace will be remembered following printing. General options for printing (including setting a standard watermark and header/footer) are available in the options under *File > Options > Personalize > Printing*.

PRINT SCHEDULE

If you want to print documents on a regular basis, e.g. an overview of important indexes every morning, you can use the print scheduling function of Tradesignal. In the schedule, you can set up the printing of any number of documents at any date and time.

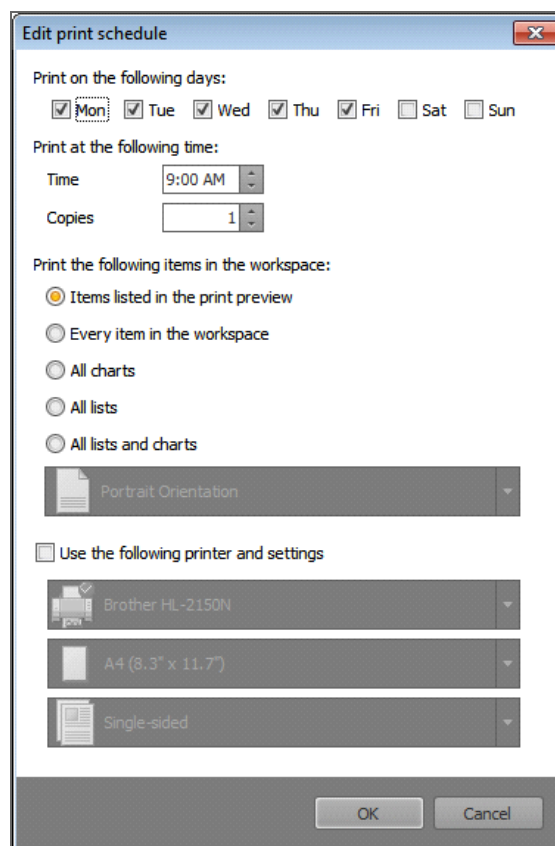
To be available for this function, the documents have to be saved as a package in workspaces. If the documents are not found there, either the wrong documents or none may be printed. Therefore, we recommend that you test each print schedule after its set-up.

To set up a print schedule

1. Open the desired workspace
2. Select **File > Print**
3. Click on **Add print schedule** at the bottom left of the print pane to display the print schedule options
4. Select the desired print schedule settings and click **OK**

The dialog will be dismissed and a new entry in the **Schedule** section of the print pane will be added. The buttons next to this entry can be used to edit or remove the schedule. Multiple schedules can be added for the same workspace.

If any workspaces have print scheduling set, a small print icon will be displayed above the Toolbar. Clicking on this button will display a dialog showing all scheduled printing jobs and provide buttons to edit or delete them.



Print Schedule

OPTIONS FOR SCHEDULED PRINTING

When the print schedule settings dialog is displayed, the following options are available. Where there is no option for printing the setting chosen from the workspace print pane will be used.

- **Days** - Which days of the week to print on.
- **Time** - At which time to print. To print at multiple times on the same day, add a second print schedule.
- **Copies** - How many copies of the document to print.
- **Items to print** - This option gives various ways to print different items in a workspace
 - Items listed in the normal print settings
 - All items in the workspace
 - All charts in the workspace
 - All lists in the workspace (Watchlists, etc.)
 - All lists and all charts
- **Orientation** - When not using the normal print settings, it is also possible to set the orientation for printing.
- **Alternate printer** - It is possible to use either the printer settings from the normal print page, or to select an alternate

printer, page size and duplex mode.

Click **OK** to save the print schedule settings.

PUBLISHING WITH THE WINDOWS CHART VIEWER

If you are often confronted with the task of creating publications with chart analysis, you can ease the work by using Tradesignal documents as Chart Viewer Object objects. This is a *Windows Object* that can be used in many programs, e.g. word processing.

In the following paragraphs, you can find the step-by-step instructions for MS Word (up to Word 2003) and MS Word 2007.

MS WORD (UP TO WORD 2003)

1. Open your word processor.
 2. From the **Insert** menu, select the **Object** entry.
 3. From the list of objects, select **Tradesignal Chart Viewer**.
 4. Right-click on the newly appearing border in the text document. The context menu opens.
 5. From the context menu, select **Tradesignal Chart Viewer Object** and then select **Properties** from the submenu to edit those.
- Here you have to enter the package in which the requested workspace is saved. For example:
 - Package: *Tradesignal Files*
 - Workspace: *Chart Nasdaq*

You can edit the format of the graph by editing the height and width. It is possible to enter more than one chart in the new document.

Save your document, e.g. as a template for your analysis. The name and folder can be freely chosen.

When you open the document again, the following steps are necessary to update the graph.

1. Start Tradesignal.
2. Right-click in the graph in your document (e.g. in MS Word) to open the context menu.
3. In the context menu, select the **Properties** of the **Chart Viewer**.
4. Click on **Update** and then on **OK**.

The chart is now refreshed with the data given by Tradesignal.

MS WORD 2007

When using MS Word 2007, the "Developer" tab has to be available. If this is not the case, proceed as follows:

1. Click on the Microsoft Office button and then click on **[program] Options**. The program is the name of the current program, i.e. "Word".
2. Click on **Popular** and check **Show Developer Tab in the Ribbon**.
3. Click on the tab **Developer**.

4. In the controls, select **Legacytools**. Then, under **More Controls**, select **TradeSignal Chart Viewer**.

In the configuration window, enter the package in which the requested workspace is saved.

For example:

- Package: *Tradesignal Files*
- Workspace: *Chart Nasdaq*

You can edit the format of the graph by editing the height and width. It is possible to enter more than one chart in the new document.

Save your document, e.g. as a template for your analysis. The name and folder can be freely chosen.

When you open the document again, the following steps are necessary to update the graph.

1. Start Tradesignal.
2. Right-click in the graph in your document (e.g. in MS Word) to open the context menu.
3. In the context menu, select the **Properties** of the **Chart Viewer**.
4. Click on **Update** and then on **OK**.

The chart is now refreshed with the data given by Tradesignal.

MS WORD 2010+

When using MS Word 2010 or above, the "Developer" tab has to be available. If this is not the case, proceed as follows:

1. Click on the **File** tab on the toolbar and then click on **Options**.
2. Click on **Customize Ribbon** and in the box on the far right that lists all ribbon tabs, check the option labelled **Developer**, then click **OK**.
3. Click on the tab **Developer**.
4. Ensure design mode is active by clicking on *Design Mode* button in the **Controls** group if it is not highlighted.
5. In the **Controls** group, select **Legacy Tools**. Then, under **More Controls**, select **Tradesignal Chart Viewer**. The chart viewer object will be inserted into your document.
6. Right-click on the inserted object and select *Tradesignal Chart Viewer Object > Edit* to show the properties dialog.

In the properties dialog, enter the package in which the requested workspace is saved.

For example:

- Package: *Tradesignal Files*
- Workspace: *Chart Nasdaq*

You can edit the format of the graph by editing the height and width. It is possible to enter more than one chart in the new document.

Save your document, e.g. as a template for your analysis. The name and folder can be freely chosen.

When you open the document again, the following steps are necessary to update the graph.

1. Start Tradesignal.

2. Right-click in the graph in your document (e.g. in MS Word) to open the context menu.
3. In the context menu, select the **Properties** of the **Chart Viewer**.
4. Click on **Update** and then on **OK**.

The chart is now refreshed with the data given by Tradesignal.

Please note: ActiveX Objects and Add-Ins from third-parties, will generally not work with 64-bit versions of Microsoft Office.

SAVE & SEND

SENDING A WORKSPACE BY EMAIL

You can send workspaces via Email right out of the Tradesignal.

1. Click on the **File > Save & Send > Send Using Email** button. The E-Mail wizard opens.
2. Select whether to include a preview image of the workspace.
3. Click **Send Email** to open a new email and attach the workspace.



Sending Workspaces by Email

SENDING EQUILLA SCRIPTS VIA EMAIL

To send an Equilla script and its dependent function files via email

1. Open the desired Equilla script.
 2. Select **Editor > Export > Send by Email** on the Toolbar
- The Email client with attached portable script file and a code preview will be opened.

IMPORTING WORKSPACES OR EQUILLA CODE FROM AN EMAIL

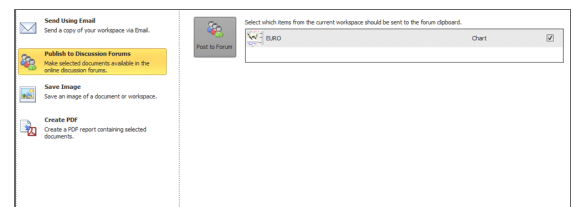
When you received an E-mail with Tradesignal documents, you can import them into Tradesignal.

1. Save the E-mail attachment in a folder that is also a package in Tradesignal.
2. In Tradesignal, open the toolbox.
3. With packages displayed, right-click on the package and select **Refresh** from the context menu.
4. When the display is refreshed, you can find the new documents in the package. You can then open them in Tradesignal.

POSTING DOCUMENTS IN INTERNET FORUMS

You can post documents from Tradesignal right into the Internet forum. To do so, you have to enter your user name and password in the Options under **File > Options > Personalize > Tradesignal Online**.

CREATE A COPY FOR THE INTERNET



Copying Documents to Forums

1. Click on **File > Save & Send > Publish to Discussion Forums**. The Publish wizard opens.
 2. Check the boxes of the items in the current workspace that you want to post.
 3. Click **Post to Forum**
1. After the copying, you can open the forum either in the current or in a new workspace. Alternatively, you can also click on **Close** and open the forum in a new web browser. In all cases, the content of the Tradesignal Online clipboard is preserved.

USE THE COPY

To use the copied document(s), you have to be signed on to the website tradesignalonline.com. Then surf to the thread in which you want to publish your document.

1. Select either **New Posting** or reply to an existing one. In the lower area of the window, a field for text entry is displayed. To its right, you can find the contents of your Tradesignal Online **Clipboard**.
2. Click on the contents to insert it in the posting. In the text field, an entry appears such as "[EQU=2132867]", referencing the contents. This unique entry cannot be edited further.

IMPORTING CHARTS, WORKSPACES OR EQUILLA CODE FROM AN INTERNET FORUM

You can download charts, workspaces and Equilla Codes from the internet forum for use in Tradesignal.

1. In Tradesignal, select **File > Help > Discussion Forums**.
2. Surf to a thread or posting that includes a Tradesignal document.
3. Click on the link **Import into Tradesignal**. If a chart was created with Tradesignal Online, you can also click on it to open it in Tradesignal. You can find more information on importing code in the chapter Downloading an Indicator from Tradesignal Online.

SAVING DOCUMENTS AS IMAGES

You can save the workspace as image, e.g. to use it later in text documents or E-Mails.

Click on the **File > Save & Send > Save Image**. The **Save Image** pane opens. At the bottom of the pane you will see a preview of the image.

OPTIONS FOR SAVING AS IMAGE

- Pick what you would like to save an image of, the entire workspace or a single element in it..
- Enter an image size in pixels, inches or millimeters. Take care that the freely adjustable height and width of the image do not lead to distortion.
- Select the **Color Mode**, either **Color** (color image with 24 bit RGB and alpha channel) or **Black and White** (no colors).
- Select the resolution for the image. 96 DPI (dots per inch) is normal screen resolution, use higher resolutions if the image is intended for print output or zooming out using interpolation is intended.
- Select whether to use interpolation. Interpolation will essentially render the image at the selected resolution, then scale it down to fit the desired dimensions. This will create a zoomed-out look to an image at the cost of some precision.

Finally, either click on of the following buttons:



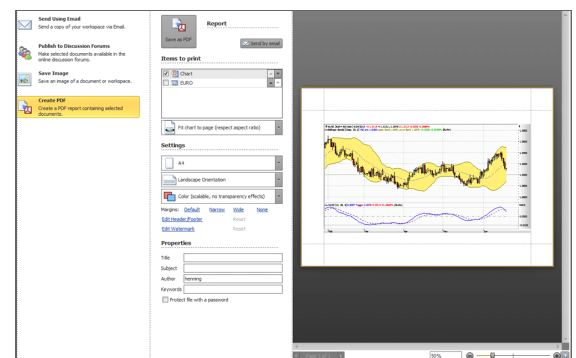
Saving Documents and Workspaces as Images

- **Save Image** - This saves the selected item as an image file. A new dialog opens. Choose the target folder and the file format. Six image formats are available: jpg, png, bmp, gif, tif and emf (emf is a scalable, vector format, although the generated image may not be identical to the preview).
- **Copy to Clipboard** - This generates a copy of the image in the Windows clipboard in bmp format.
- **Send Email** - Create the image in png format and attach it to a new email.

SAVING PDF DOCUMENTS

Any items that can be printed can also be saved as a PDF file. To save a PDF file click on *File > Save & Send > Create PDF* to display the PDF preview pane.

The PDF preview pane present most of the same options that the print preview pane offers, and uses previously saved settings associated with a workspace.



PDF Wizard

- **Save as PDF** - Save the PDF file.
- **Send by Email** - Create a PDF file and attach it to a new Email.
- **Items to Print** - List of items in the workspace to print (including an extra page containing the workspace as a whole. The items are printed in the order listed. The arrow buttons can be used to change the print order of the items.
- **Chart Scaling** - Determines how charts should be scaled to fit the printed page.
- **Paper Size** - Selection of the paper size.
- **Orientation** - Portrait or landscape.
- **Color Mode** - Select whether the document should be printed in **Full Color (with transparency)** (24 bit RGB with alpha channel), **Full Color (without transparency)** (24 bit RGB without alpha channel) or as **Black and White**. If the image is scaled on your printer (e.g. into a larger format), select **Full Color (without transparency)** for better results.
- **Margins** - Some predefined margin options. Margins can also be manually adjusted using click and drag directly in the preview pane.
- **Edit Header/Footer** - Shows the header/footer dialog where text and various macros can be set for the page header and footer. The alignment of text and font can also be changed. The following macros are available:
 - Document title
 - Page Number
 - Page Count
 - Print Date
 - Print Time
 - Print User

- **Edit Watermark** - Show the watermark dialog where a watermark based on text or an image can be specified for each page on the printed output. The watermark dialog presents many options to configure how the watermark is displayed, including
 - Text, Text direction, Font, Color and Size.
 - Image file and alignment.
 - Translucency.
 - Positioned in front or behind.
 - Page range to apply watermark to.
- **Document Properties** - Title, subject, author and keywords meta data that will be embedded in the PDF file.
- **Protect file with a password** - When checked and the PDF is saved, a dialog will be displayed where the following passwords can be set
 - Password to open the PDF document
 - Password to edit the PDF document
 - Restrictions on printing
 - Restrictions on editing
 - Restrictions on copy portions of the file

COPYING DOCUMENTS VIA THE WINDOWS CLIPBOARD

If you want to transfer graphs and tables quickly to other documents, for example to further analyze the data, you can use the Windows clipboard. This way, you can copy a chart into a text document or copy the results of the Optimizer into a table. This copying also works with other tables or with Equilla codes. Instead of using the key combinations, you can also use the buttons in the *Edit* group of the toolbar.

COPYING CHARTS

1. In an active chart, press the key combination **Ctrl + C** to copy the chart.
2. Switch to the target document, e.g. publishing software.
3. There, press the key combination **Ctrl + V** to paste the chart.

The chart is inserted in bitmap format.

If the target document is a table, for example in MS Excel, you can copy the price data as well as any indicators or strategies included in the chart into the Excel table. To do so, select the price data in Tradesignal and then drag the data into the Excel table with the mouse.

COPYING TABLES

1. Select the rows in the source table that you want to copy (the source may be a Scanner, Watchlist, Portfolio or Optimizer).
To select all rows, press **Ctrl + A**.
2. Press **Ctrl + C** to copy the rows.
3. Switch to the target document, for example a Microsoft Excel table.

4. Press **Ctrl + V** to paste the table contents into the target document.

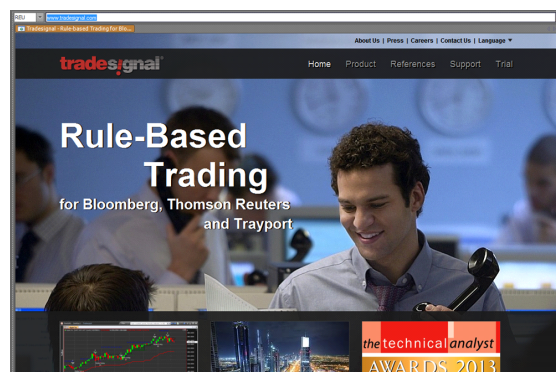
COPYING TEXT

1. Select the lines that you want to copy, either in an Equilla Editor, an HTML Editor or a News window. To select all lines, press **Ctrl + A**.
2. Press **Ctrl + C** to copy the lines.
3. Switch to the target document, for example a text editor, and press **Ctrl + V** to paste the contents.

WEB BROWSER

The scope of the stock market is not limited to one's own office. Information from all over the world has to be gathered and exchanged with other users or colleagues. To do so, Tradesignal offers you an integrated web browser. This way, you have access to the web contents of tradesignalonline.com and every other web page in the Internet. Use the web browser for

- surfing the Internet
- communicating in Internet forums
- contacting the support staff
- displaying web-based help contents



Web browser

Tradesignal also offers an HTML Editor as a language interface to JavaScript. With it, many functions of the Tradesignal software can be controlled. You can program your own tools to make working with Tradesignal easier and faster.

For information about distributing content on the Internet, for example in forums, please refer to the chapter Publishing.

USING THE WEB BROWSER

In the *Insert* tab in the toolbar, click on the button **Web Browser**. An empty web browser opens.

You can also open a web browser by entering a URL in the command line and selecting **New Browser** from the menu.

To replace an existing website, enter the URL in the command line and select **Replace Web Page** from the menu.

To navigate back and forth between sites, click on the small green arrows to the right of the entry field in the command line.

To close a browser window, either press **Ctrl + F4** or open the context menu of the tab and select the entry **Close [window name...]**.

HTML EDITOR

Tradesignal possesses a built-in HTML editor. With it, you can do the following:

- Use HTML code to design a GUI or control elements like buttons, lists or entry fields
- Use active elements written in JavaScript (JS), for example to list all active workspaces. This way you can write control elements that use program functions of Tradesignal.

USE THE HTML EDITOR

Click on the **Insert** tab on the toolbar then choose the entry **HTML Page**.

An editor window opens. In the toolbar, the *Editor* tab appears with the following buttons:

Show Browser - Interprets the HTML or JavaScript Code and displays the result in a browser.

Show Editor - Switch from the browser back to the editor.

Undo - Undo the last action (multiple undoes possible).

Redo - Repeat the last action (multiple repeats possible).

Select All (Ctrl+A) - Selects all code. You can use the standard Windows key commands to copy the code with "Ctrl+C" and then paste with "Ctrl+V", e.g. into a text editor.

To display HTML or JS code, simply enter it in the HTML editor and click on **Show Browser**.

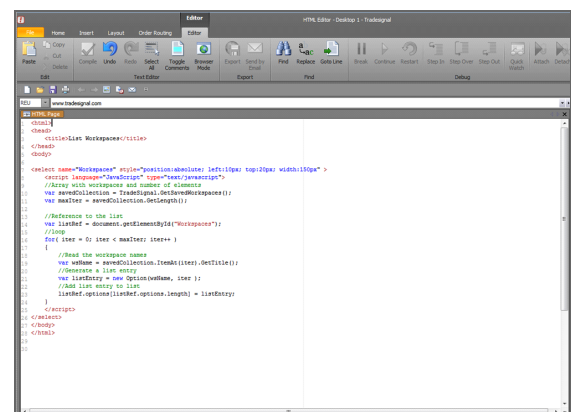
SEARCH AND REPLACE

For editing the code, the *Find* group in the toolbar is available.

- Click on **Find** to enter a search term in a search pane.
- Click on **Comment/Uncomment Selection** to comment/uncomment selected lines using HTML comments.
- With **Replace** you can replace one or all appearances of a code text with another text.
- For very long scripts, you can use **Goto Line** to jump to a code line.

JAVASCRIPT IN THE HTML EDITOR

JavaScript offers you a connection to the inner workings of Tradesignal. With JS and a number of interface commands it is possible to control various functions of Tradesignal. Here a list of possible script commands:



HTML Editor

```
// Global variable to access the TSEPublic object, e.g.:
//
// var selectedWorkspace = TradeSignal.GetSelectedWorkspace();
//
TradeSignal
// The Tradesignal script root element, reachable with JavaScript by either using window.external
class TSEPublic
    // Returns a collection of all saved workspaces
    GetSavedWorkspaces() : TSEWorkspaceCollection

    // Returns a collection of all open workspaces
    GetOpenWorkspaces() : TSEWorkspaceCollection
    // Returns the currently selected workspaces
    GetSelectedWorkspace() : TSEWorkspace
    // Returns the last selected workspace, selected before the
    // current workspace was selected, this is useful for operating
    // on a target workspace from within an HTML page in another
    // desktop
    GetLastSelectedWorkspace() : TSEWorkspace
    // Returns the total RAM managed and used by Tradesignal
    GetTotalMemory() : int
    // Returns the current timezone set by the user in
    // Olsen format (e.g. Europe/Berlin)
    GetUserTimeZone() : string
    // Returns the current version number of Tradesignal (x.y.z)
    GetVersion() : string

    // Like the Bang! Command !new
    Create( whatToCreate : string, parameters : string ) : TSEDocument
    // Opens a new chart and returns it with an instrument
    CreateNewChart( symbol : string ) : TSEDocument
    // Opens a new chart with an instrument optionally allowing
    // the chart type and scale alignment to be set.
    // Use a parameter string of the form:
    //
    // "charttype=line(open); scale=left; subchart=false"
    //
    // valid values for scale are left|right|both
    // valid values for charttype are:
    //   line(field), bar, candlestick, candlevolume,
    //   equivolume, symbol(field), stepped(field),
    //   area(field1, field2), forest(field), linkedforest(field),
    //   renko(field), pointfigure, heikinashi, kagi, tlb
    // valid values for subchart are true|false
```

```
// valid values for field are open|close|high|low|volume|openinterest
CreateNewChartEx(symbol : string, parameters : string) : TSEDocument
// Opens a new Market Profile and returns it with an instrument
CreateNewMarketProfile( symbol : string ) : TSEDocument
// Opens a new chart and returns it with an indicator
CreateNewInstantIndicator( equillaCode : string ) : TSEDocument
// Opens a new chart and returns it with an indicator series
CreateNewInstantIndicatorSeries( equillaCode : string ) : TSEDocument
// Opens and returns a new webbrowser containing the given url
CreateNewBrowser( url : string ) : TSEDocument
// Opens and returns a new scanner with an instrument
CreateNewMarketScanner( symbol : string ) : TSEDocument
// Opens and returns a new workspace
CreateNewWorkspace() : TSEWorkspace
// Passes the entered key string as the value of an additional
// parameter called sessionkey that will be appended to all
// TXML data requests to a server identified by the URL
// parameter
SetMixinDataSessionKey(url : string, key : string)
// Shows the Tradesignal Help
ShowHelp()
// Shows the Equilla Help
ShowEquillaHelp()
// A collection of workspaces. The collection is zero (0) based
class TSEWorkspaceCollection
    // Returns the workspace of a certain position in the collection
    ItemAt( index : int ) : TSEWorkspace

    // Returns the number of workspaces in the collection
    GetLength() : int

    // Returns the workspace with a certain name
    Find( workspaceName : string ) : TSEWorkspace
// An element representing a certain workspace in Tradesignal
class TSEWorkspace
    // Returns the workspace title
    GetTitle() : string
    // Returns true in case the workspace is open, otherwise false
    IsOpen() : bool
    // Returns true in case the workspace is included in the currently open file list, otherwise false
    IsRecent() : bool
    // Returns true if the workspace was open at the last shutdown of Tradesignal, otherwise false
    WasPreviouslyOpen() : bool
    // Opens the workspace or activates it, if it is already open
    Open() : bool
```

```
// Closes the workspace; optionally other saving dialogs can be suppressed
Close( suppressDialogs : bool ) : bool

// Like the Bang Command !new except that it selects the workspace first
Create( whatToCreate : string, parameters : string ) : bool
// Selects the workspace if it is open
Select() : bool
// Returns a collection of all documents in the workspace
GetDocuments() : TSEDocumentCollection
// Returns the currently selected document in the workspace
GetSelectedDocument() : TSEDocument
// A collection of workspace elements (documents), the collection is zero (0)
// based
class TSEDocumentCollection
    // Returns the document of a certain position in the collection
   ItemAt( index : int ) : TSEDocument
    // Returns the number of documents in the collection
    GetLength() : int
    // Returns the document with a certain name
    Find( workspaceName : string ) : TSEDocument
// An object which represents an element in the workspace, this could be
// a Chart, Browser, Scanner, etc.
class TSEDocument
    // Returns the document name
    GetDocumentTitle() : string
    // Returns the name of the selected element in the document
    GetSelectedItemTitle() : string

    // Returns the document type: Chart, Browser, TextEditor, PriceEditor,
    //MarketProfile, MarketScanner, WorkspaceInterfaceScript, Statistics, Trades or Optimiser
    GetDocumentType() : string
    // Selects the document
    Select() : bool
    // Selects the document and executes !replace
    Replace( itemToReplaceSelectedItem : string ) : bool
    // Selects the document and executes !replaceall
    ReplaceAll( itemToReplaceAllCurrentItems : string ) : bool
    // Selects the document and executes !add
    Add( itemToAdd : string ) : bool
    // Selects the document and adds an item optionally allowing
    // the chart type, scale alignment and subchart to be set.
    // Use a parameter string of
    // the form:
    //
    // "charttype=line(open); scale=left; subchart=false"
```

```
//
// valid values for scale are left|right|both
// valid values for charttype are:
//   line(field), bar, candlestick, candlevolume,
//   equivolume, symbol(field), stepped(field),
//   area(field1, field2), forest(field), linkedforest(field),
//   renko(field), pointfigure, heikinashi, kagi, tlb
// valid values for subchart are true|false
// valid values for field are open|close|high|low|volume|openinterest
AddEx( itemToAdd : string, parameters : string ) : bool
// Selects the document and executes a setp property action,
// valid property names and values are:
//
//   "period" - period in seconds (0 for tick, 86400 for daily)
//   "dateaxismode" - normal|seasonal
//   "historylength" - number of bars of data to load
Set( propertyName : string, value : string ) : bool
// Selects the document and executes sets the value for all
// instruments in all subcharts, see Set() for available
// properties
SetAll (propertyName : string, value : string) : bool
// Evaluates the Equilla Code and adds it to the document
Eval( equillaScript : string ) : bool
// Evaluates the Equilla Code Series and adds it to the document
EvalSeries( equillaScript : string ) : bool
// Selects the document and executes !close document
Close() : bool
```

EXAMPLE: LISTING ALL AVAILABLE WORKSPACES WITH JAVASCRIPT

In this tutorial we will design a simple JavaScript that will display a list of all available workspaces.

STEP 1: START A HTML PAGE AND ENTER THE PAGE FRAME

Open a new, empty workspace. Then open the button menu **Layout** and select the entry **Insert HTML Element**. An empty HTML editor opens. For a correct page frame, please enter the following source code:

```
<html>
<head>
  <title>List Workspaces</title>
</head>
<body>
</body>
```

```
</html>
```

STEP 2: ADDING THE LIST ELEMENT AND THE SCRIPTING AREA

Now copy the following line between the **Body** Tags:

```
<select name="Workspaces" style="position:absolute; left:10px; top:20px; width:150px" >
</select>
```

The line includes the HTML code for the list with position and width values. When you now click on **Show Browser** in the toolbar, you can see the new page with an empty list. Click on **Show Editor** to edit the code again.

Now we can define the area for the JavaScript source code. Copy the following code into the line below the list. The code assigns the collection of all saved workspaces to a variable.

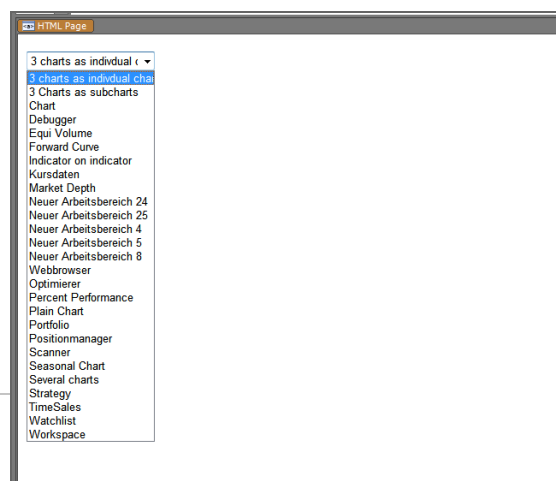
```
<script language="JavaScript" type="text/javascript">
//Array with workspaces and number of elements
var savedCollection = TradeSignal.GetSavedWorkspaces();
var maxIter = savedCollection.GetLength();
</script>
```

STEP 3: DESIGNING AND TESTING THE JAVASCRIPT

In the upper part we have set up a HTML page and the basis for a JavaScript. The list of workspaces was copied into a variable. To be able to read out the list later, we need the number of included elements. This value is also saved as a variable.

In the following script, a reference to the list is set, so that we can add a list entry for every workspace in the collection. To read out the collection, we need a loop that runs over all workspaces. Add the following lines in the script area:

```
//Reference to the list
var listRef = document.getElementById("Workspaces");
//loop
for( iter = 0; iter < maxIter; iter++ )
{
    //Read the workspace names
    var wsName = savedCollection.ItemAt(iter).GetTitle();
```



Final script in the editor

```
//Generate a list entry
var listEntry = new Option(wsName, iter );
//Add the list entry to the list
listRef.options[listRef.options.length] = listEntry;
}
```

THE COMPLETE SCRIPT

Here you can see the complete code of the script.

```
<html>
<head>
  <title>List Workspaces</title>
</head>
<body>
<select name="Workspaces" style="position:absolute; left:10px; top:20px; width:150px" >
<script language="JavaScript" type="text/javascript">
//Array with workspaces and number of elements
var savedCollection = TradeSignal.GetSavedWorkspaces();
var maxIter = savedCollection.GetLength();
//Reference to the list
var listRef = document.getElementById("Workspaces");
//loop
for( iter = 0; iter < maxIter; iter++ )
{
  //Read the workspace names
  var wsName = savedCollection.ItemAt(iter).GetTitle();
  //Generate a list entry
  var listEntry = new Option(wsName, iter );
  //Add list entry to list
  listRef.options[listRef.options.length] = listEntry;
}
</script>
</select>
</body>
</html>
```

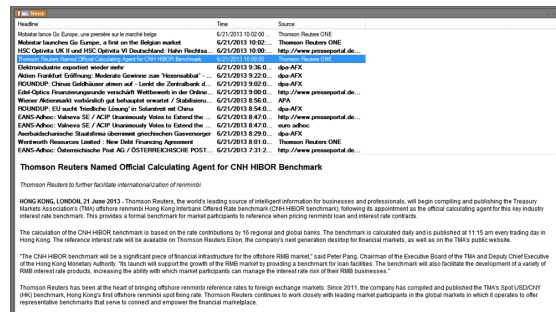
To view the result, click on **Show Browser** again. A list of all workspaces appears.

Note that you cannot save the HTML script itself, only the workspace in which it was created.

NEWS

News is an important tool for the fundamental chartist to have at their disposal and can be accessed from Tradesignal Standard Edition either from a traditional news reader or as a study within a chart. Tradesignal allows you to view both realtime news from your data feed and also to subscribe to RSS sources available via the Internet or a local intranet.

The Tradesignal news reader uses a split display to show recent headlines in the top pane, and the selected story in the bottom pane. The window sizes can easily be adjusted by dragging the separator between the two panes.



The news reader

Use the news tools to:

- View realtime news as it happens
- View current news for a specific security
- View historic news in your charts
- Connect to RSS feeds to view any event-based information

NEWS SETUP

You can view news stories in one of the following ways:

1. To view all recent news headlines, click on the **News** button in the *Insert* tab of the toolbar.
2. To view all headlines for a specific security, right-click on a security in a chart, watchlist, portfolio, scanner or symbol list to show the context menu, then select the option **Open**, entry **News**.
3. To view news stories in a chart, drag & drop the **News** indicator from the toolbox onto a security in your chart.

LINKING THE NEWS READER TO A WATCHLIST OR CHART

If you want to link a news reader so that it displays relevant news depending on what is currently selected in a watchlist or chart, do the following:

1. Right-click on the tab above the chart or watchlist and pick a color under the *Symbol Link* menu.
2. Click on **News** in the *Insert* tab of the toolbar to display a news reader.
3. Right-click on the tab above the news reader and pick the same color link you chose in the chart or watchlist.

The headline list will now only show stories related to the selected item in the watchlist or chart. To change the displayed headline, just select a new item in your watchlist, or replace the item in your chart.

WORKING WITH NEWS IN A CHART

By applying the news indicator to a security in a chart as described above, you will get a historic overview of headlines related to that security displayed as hyperlinks above the bar on which the news was published. Just click on one of the hyperlinks to open a news reader in which you will see the full news story.

The news indicator is written in Equilla using the news API. Programmers can make use of this API to create their own indicators or indeed trading strategies based around realtime news. Please refer to the Equilla help system within Tradesignal for more information on this API.

FORMATTING OPTIONS

The following settings can be used to customize the news window to your preference:

Back color - Change the background color for news headlines and stories.

Text color - Change the text color.

Font size - Change the size of the text in headlines and stories.

Show time - Displays the time of the news story in the headline list.

Show source - Displays the source (feed) of the news story in the headline list.

Browser alignment - Display the headline list to the top, right, bottom or left of the news story pane.

FILTER OPTIONS

Show - *All* available news headlines; Headlines for a selected *symbol*; *Stories* for a given symbol on a specific day.

Symbol - Displays the currently selected security for which headlines are shown.

Days - How many days of headlines should be displayed.

Keywords - Show only headlines containing the entered keywords. **RSS Feeds** - Displays how frequently (in seconds) the headlines from RSS feeds should be updated.

Note: You can save these settings as default by clicking on the small disk icon above the settings in the property inspector. Alternatively, you can create a news reader style by selecting *Home > Style > Styles > Create Style* from the *Format* group on the toolbar.

FUNCTIONS, INDICATORS AND STRATEGIES

WORKING WITH EQUILLA

EQUILLA BASICS

With the programming language Equilla, Tradesignal offers you an easy to learn but powerful programming language. With its help, you can easily perform even complex tasks.

Equilla resembles programming languages like Pascal or Visual Basic. All functions, indicators and strategies in Tradesignal are written in Equilla. (For the relationship between these three, see the chapter Introduction to Functions, Indicators and Strategies).

The special benefits of Equilla are the extensions for the trading business. They allow you to translate your ideas about finance and analyses into executable routines. This way, you can write your own strategies, indicators or other applications for the various software components of Tradesignal. Equilla helps you to solve your individual problems, to test new trading ideas and to work outside the box of externally defined, inflexible procedures.

To quickly get accustomed to Equilla, we recommend familiarity with a programming language. However, you can also learn to tap the potential of Equilla by analyzing and editing available Equilla code (e.g. customizing an indicator).

Equilla scripts are edited in the Equilla Editor, which supports you with color-coding of the Equilla source code. In the editor, you can also retrieve detailed information for Equilla functions via the context menu.

The Equilla chapters in this help are intended to be more of an introduction to the various topics. For an extensive documentation of all features, please refer to Tradesignal help under the menu entry **Equilla Formula Language**.

SEQUENCE OF EVENTS WHEN COMPILING EQUILLA CODE

When compiling Equilla code into an executable Equilla program, it is important to remember that all Equilla scripts are saved in packages. The compilation sequence partly depends on the sequence of these packages, which you can change in the Package Management in the Tradesignal Options.

CALLING UP FUNCTIONS FROM INDICATORS/STRATEGIES

For indicators and strategies, functions are called up by their name (without information about their package). Tradesignal searches for the function in the following sequence:

1. in the same package as the indicator/strategy
2. in all packages, in the sequence given in the Package Management in the Tradesignal Options

As soon as a function with the correct name is found, the search ends and this function is used.

Therefore, if you move indicators/strategies to another package, it may give different results if another function is used due to the changed package sequence.

Exporting packages can also lead to problems, if packages exchanged with other users use functions from another package. If the user has no function with the same name, the script will not compile; if the user has another or older version, the results for the users may differ.

CALLING UP INDICATORS/STRATEGIES FROM SAVED DOCUMENTS

When saving a chart, scanner, watchlist or portfolio, not only all settings but also all applied indicators and strategies (including their respective package names) are saved.

When the documents are opened again, Tradesignal checks whether a newer version of the indicator/strategy is available, and will use that one if possible. Tradesignal searches in the following sequence:

1. in the packages as given in the saved document
2. in all packages, in the sequence given in the Package Management in the Tradesignal Options

If no newer version is to be found, the indicator/strategy as saved in the document is used.

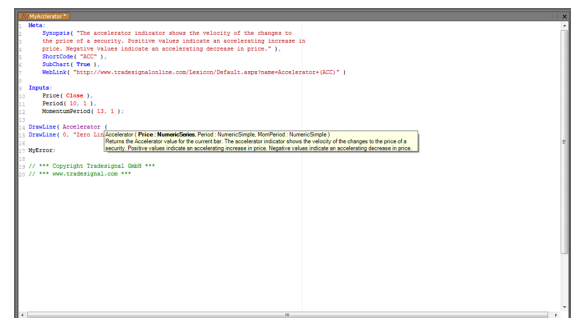
EQUILLA EDITOR

For writing and editing scripts written in the Equilla language (which is in-built in Tradesignal), an Equilla editor is available. It supports programming by highlighting code areas and comments in different colors.

OPENING THE EQUILLA EDITOR

You can open the Equilla editor in two ways:

- The editor opens when you edit a function, indicator or strategy. To do so, open the respective list (functions, indicators or strategies) in the toolbox. Then call up an entry for editing by either double-clicking on it or by selecting **Edit** from the context menu. The Equilla editor opens with the source code of the script.
- To open an empty editor, you have to start a new script. To do so, select **New Function** (or **New Indicator** or **New Strategy**) in the *Related Task* area in the toolbox. The Create Item Wizard opens. Select the option **Use the Equilla Formula Language to write the [...]**. Enter a name and select a location for saving. Finish with **Done**. An empty editor opens.



Equilla Editor

COLOR CODING IN THE EDITOR

The source code is color-coded. The following colors are possible:

Blue: Signal words in the code (begin, end), declaration areas (meta, variables...), input types, commands (if then...), formatting (StyleSolid...), drawing commands (DrawText...).

Red: Content of synopsis, numeric values (e.g. close or true numbers like 1, 10, 30), true/false.

For blue and bold red Equilla expressions, you can find additional information in Tradesignal. To display it, right-click on the operator in question in the Equilla Editor, e.g. *Begin*, and choose **Lookup Equilla Function** from the context menu. A browser window opens with information on the operators and links to related functions.

Lilac: Names of other functions.

You can open these functions in the Equilla Editor. To do so, right-click on the function and choose **Open** from the context menu. The source code is opened in the Equilla Editor.

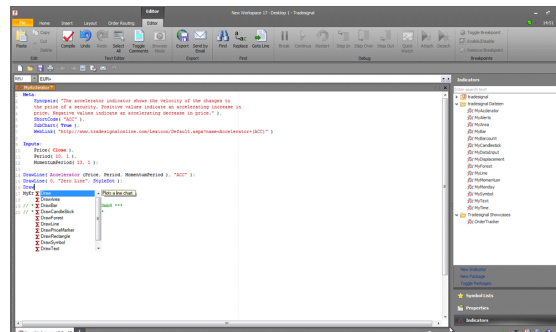
Green: Comments

Black: Anything else, like text, declared variables etc.

EDITING THE EQUILLA CODE

When the Equilla editor is open, the *Editor* tab appears in the toolbar, with the following buttons:

Compile Script (F7) - Use this to compile the script. Only compiled scripts can be used. Un-compiled scripts are grayed out in the toolbox. In case of script errors, error descriptions will be given in the lower area of the workspace. For delivered scripts, compiling may lead to error messages if the scripts are protected against overwriting.



Equilla Editor group in the toolbar

Undo (Ctrl+Z) - Undo the last action (multiple undoes possible, but none after compiling). This way, you can undo an unintended deletion, for example.

Redo (Ctrl+Shift+Z) - Repeat the last action (multiple repeats possible, but none after compiling). This way, you can insert a copied text multiple times, for example.

Select All (Ctrl+A) - Selects all code. You can use the standard Windows key commands to copy the code with "Ctrl+C" and then paste with "Ctrl+V", e.g. into a text editor.

SEARCH AND REPLACE

To edit the code, the *Find* group in the toolbar is available.

- Click on **Find** to enter a search term in the search pane.
- Click on **Comment/Uncomment Selection** to comment/uncomment selected lines of code using Equilla comments.
- With **Replace** you can replace one or all appearances of a code text with another text.
- For long scripts, you can use **Goto Line** to jump to a code line.

APPLYING YOUR EQUILLA SCRIPT

Scripts can only be applied once they have been compiled. After that, the new script can be used like any delivered function, indicator or strategy. You can find more information in the chapter Introduction to Functions, Indicators and Strategies.

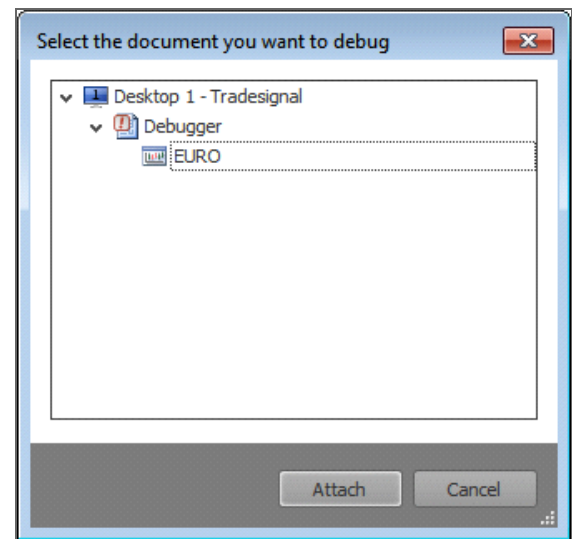
EQUILLA DEBUGGER

The Equilla debugger can be used to find and resolve problems in your Equilla scripts. It allows you to stop execution of your scripts, step through your source code and examine the state of variables used in your code.

CREATING A DEBUGGER DOCUMENT

To create a new debugger document, select the Insert group on the Toolbar and click on the Debugger button. After clicking on the Debugger button, a dialog will be displayed asking you which document you would like to debug. The dialog shows a list of all open desktops, workspaces and documents and presents the following options:

- **Skip** – Do not attach to a document yet. This is typically used if the document you want to debug isn't open yet. You may execute the debuggers Attach command from the ribbon as soon as the document you want to debug is ready to be used.
- **Attach** – Attaches the new debugger to the selected document.
- **Cancel** – Abort the operation and do not create a new debugger document.



Debugger Attach Dialog

Please be aware that a debugger can only debug one document at a time and only one debugger can be attached to a specific document. You may create multiple debuggers simultaneously and attached each of them to a separate document.

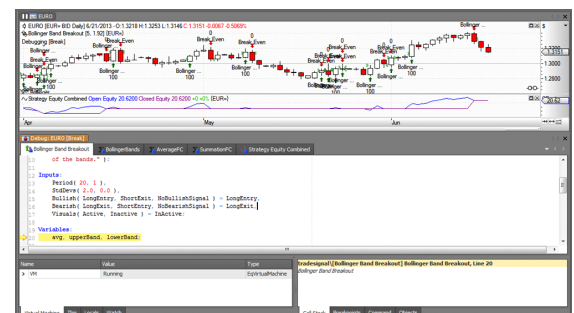
In order to be able to debug an indicator or strategy it has to be compiled using Tradesignal 7.0 or higher. Equilla scripts that have been compiled with earlier versions of Tradesignal will not show up in the debugger document.

Please note: Only chart documents can be debugged.

THE DEBUGGER DOCUMENT

The debugger document enables you to debug your Equilla scripts. It consists of the following components:

- **Scripts** – The debugger loads all scripts that are referenced in the debugged document.
- **Watch windows** – A set of watch windows that help you inspect the state of your indicator or strategy. There are three predefined windows, showing common information and one window where you can enter debug expressions on your own.
- **Call Stack** – Shows the nesting level of your scripts while



The debugger

debugging.

- **Breakpoints** – Shows a list of all breakpoints that have been set up.
- **Objects** – Provides a list of all Equilla indicators and strategies that are contained in the debugged document. You may enable or disable the debugging of specific objects as you need.

SCRIPTS

Every script referenced in the debugged document will be loaded automatically. Here you can inspect the source code, add or manipulate breakpoints or examine the state of variables as you need.

The debugger editor has all the same functionality as the normal Equilla Editor.

To add a breakpoint, click on the left margin of the editor, select the Toggle Breakpoint command from the ribbon or use the corresponding context menu entry.

While the execution is paused, you can examine the state of variables or evaluate Equilla expression by hovering over the corresponding text with your mouse. To evaluate an Equilla expression containing space characters, you have to select the whole expression in the editor before hovering over it.

Note: If a script has been changed since the indicator or strategy has been compiled, the scripts tab will show a corresponding image. Please be aware that breakpoints and variables may behave unexpected in such a case.

If the evaluation is paused, the source code line that is about to be evaluated will be highlighted in the source code editor. Additionally the margin will show an arrow icon on the corresponding line.

WATCH WINDOWS

The Equilla debugger provides a quick way to check the state of the document as a whole or of specific elements in it.

- **Virtual Machine** – This is a predefined view and cannot be modified. It shows information of all elements within the virtual machine. This includes all the indicator and strategies that have been created as well as the instruments that are used to evaluate your scripts.
- **This** – This is a predefined view and cannot be modified. It provides access to the state of the currently debugged indicator or strategy. This includes general properties as well as the referenced parents and generated output data. You may also get access to child objects that have been applied to this indicator or strategy.
- **Locals** - This is a predefined view and cannot be modified. This view provides access to all variables that are used in the currently debugged indicator, strategy or Equilla function. While stepping through your code, this view will be updated automatically to show only the variables that are relevant for the script that is about to be evaluated.
- **Watch** - Use this view to track specific variables or Equilla expressions. The list of expressions will not change automatically while stepping through your code. To add an expression, simply type it into the list or use drag and drop to copy an expression from the source code editor or another watch window. The **Quick Watch** is an alternative to this view

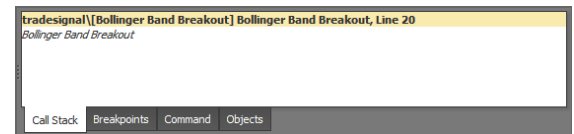
Name	Value	Type
avg	1.33236	double
Bearish	LongExit (0)	int
Bullish	LongEntry (0)	int
Close	1.3151	Close

Locals watch window

when the information for the inspected object or variable does not fit into the available space.

CALL STACK

The call stack shows the currently evaluated script and line number. The currently evaluated script is displayed on the top and the indicator or strategy script is displayed at the bottom.



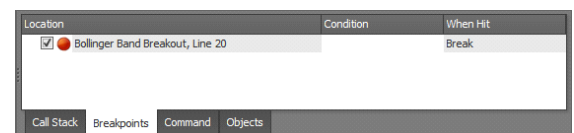
Call stack

For Equilla functions, the passed parameters will be displayed as well. Please be aware that the same limitations as for the evaluation of debug expression apply. Therefore the list of parameters might not be complete.

To navigate through the call stack, simple double click on an entry in the list.

BREAKPOINTS

This view shows a list of all the breakpoints that have been set up for a debugger. The displayed information includes the script file and line number, the breakpoint belongs to, as well as information about the state, conditions and the action that should be taken when the breakpoint is hit.



Breakpoints

State	
	The breakpoint is valid and enabled.
	The breakpoint is valid, but currently disabled.
	The breakpoint cannot be mapped to any executable code.

To modify an existing breakpoint, either enable or disable it using the check box at the front of the entry or right click on an entry to show up a context menu. The context menu provides access to the following commands:

- **Enable/Disable breakpoint** - Enable or disable the selected breakpoint
- **Remove Breakpoint** - Delete the selected breakpoint
- **Goto** - Go to the source code line the breakpoint is attached to.
- **Properties** - Modify the breakpoint condition and action.

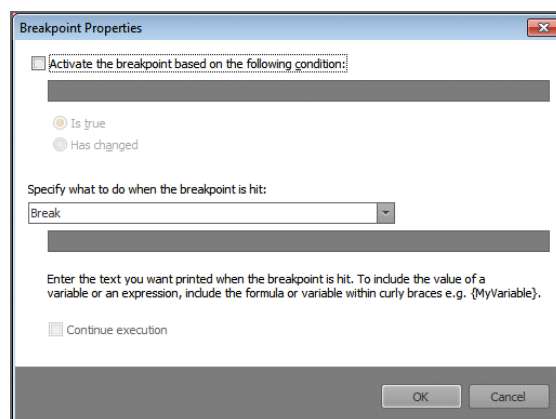
BREAKPOINT PROPERTIES

Every breakpoint supports a range of properties to control its behavior. To examine or modify these properties, right click on a breakpoint and select **Properties** from the context menu.

Breakpoints can be configured to only pause evaluation under a certain condition. To make a breakpoint conditional, check the **Active the breakpoint based on the following condition** option. Type the condition as Equilla code in the corresponding editor and specify how to interpret the condition.

Beside conditions, you can also specify what to do when the break point is hit. The default behavior is to pause evaluation, but you can also specify that the debugger should simply print a message to the output window. Available options are:

- **Break** – Pauses the evaluation when the breakpoint is hit. This is the default behavior.
- **Print a message on historic bars** – Prints the specified message when the breakpoint is hit while evaluating historic bars.
- **Print a message on real-time bars** – Prints the specified message when the breakpoint is hit while evaluating real-time bars.
- **Print a message on every bar** – Always prints the specified message when the breakpoint is hit.
- **Continue execution** – Specifies if the debugger should continue execution when the breakpoint is hit or if the evaluation should be paused as it is done for standard breakpoints.



Breakpoint Properties Dialog

BREAKING ON RUNTIME ERRORS

By default, if a runtime error occurs, the debugger will automatically break on the instruction that caused the error. This allows the developer to identify exactly what went wrong, without having to manually set breakpoints to stop just before the error.

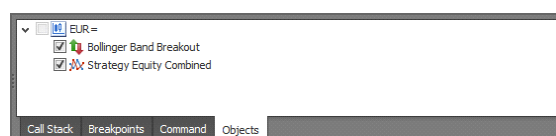
This feature may be enabled or disabled in the property inspector, using the *Break on Runtime Error* property.

In the case that runtime errors are disabled in a Equilla Script, the break point will still be triggered, although the execution can then be resumed afterwards.

OBJECTS

The objects view shows what objects are available for debugging. It also enables you to filter what items should be debugged.

To enable or disable debugging for an object, simply set or clear the check box in front of it.



Objects view

Use the context menu to toggle between these two modes.

DEBUGGER COMMANDS

The debugger document supports the following commands:

DEBUG GROUP

- **Break** - Pauses the execution when the next evaluation starts. Please be aware that some new data has to arrive in order to trigger an evaluation. If the market is closed, the debugger will pause with the next tick that arrives.
- **Continue** – Continues evaluation when the evaluation is paused.
- **Restart** - Restarts the evaluation of the complete chart. All scripts will be reevaluated starting with bar 1.
- **Step In** – Steps into an Equilla script function, entering the next level in the call stack. If the next instruction is not an Equilla script function, this command acts like the Step Over command.
- **Step Over** – Steps over the next line.
- **Step Out** – Steps out of the current Equilla script, going back one level in the call stack.
- **Quick Watch** – Shows the quick watch dialog that can be used to evaluate Equilla expressions or inspect the state of variables and objects.
- **Attach** – Attaches the debugger to a document.
- **Detach** – Detaches the debugger from the currently debugged document.

BREAKPOINTS GROUP

- **Toggle Breakpoint** – Adds a new or removes an existing breakpoint.
- **Enable/Disable** – Enables or disables an existing breakpoint.
- **Remove Breakpoint** – Removes an existing breakpoint.

PROPERTIES

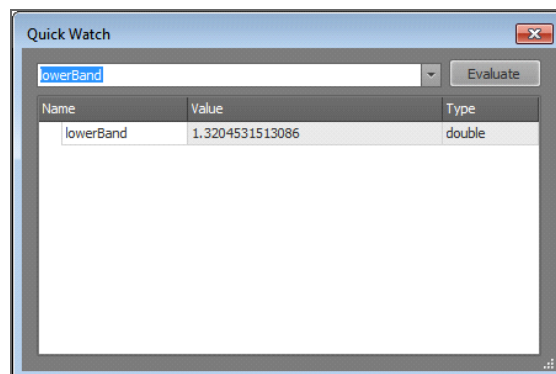
The debugger document offers the following properties to be modified from the Property Inspector:

- **FontSize** – Controls the size of the text in the Equilla script views. The default value is 9.
- **Trace Order State Changes (Historic)** – Enables or disables the tracing of order state changes while evaluating historic data. The default value is false.
- **Trace Order State Changes (Real-time)** – Enables or disables the tracing of order state changes while evaluating real-time bars. The default value is false.

• QUICK WATCH

The Quick Watch can be used to inspect the state of variables and object and to evaluate Equilla expressions.

While providing the same functionality as the Watch view it allows the resizing of the output area without affecting the general layout of the debugger document. Therefore it can be used to quickly inspect the state of a variable or object even if the information does not fit into the space available for the Watch view.



Quick Watch Dialog

DEBUG EXPRESSIONS

The Equilla debugger supports a wide range of debug expressions that can be evaluated in the **Watch** view or the **Quick Watch** dialog. This includes some predefined expressions that provide information about the debugged document or the currently debugged object.

PREDEFINED EXPRESSIONS

- **VM** – This expression provides access to the state of the debugged document. This includes settings, references to all instruments used in the document as well as all elements in the chart, like instruments, indicators, strategies and tools. When debugging strategies it also provides information about positions, orders and statistics.
- **This** – This expression provides access to the state of the currently debugged object. This includes parent objects and input instruments, as well as, error information, generated output data and trading related data.

EQUILLA EXPRESSIONS

In addition to the predefined debug expressions, you can evaluate standard Equilla code and evaluate it in the debugger. While you can use all build-in Equilla functions, you cannot use any Equilla script function.

Valid examples, using only build-in functions or standard operators:

```
Sum(close, 10) / 10
(Close[1] + Close) / 2
```

Invalid example, using a script function rather than a build in function:

```
Average(Close, 10)
```

Please be aware that some variable might not be available in the debugger. This is caused by code optimizations that are applied to the source code when compiling it. This may result in variables being removed or replaced in order to save resources and

increase the evaluation speed. Additionally the compiler may reorder the instructions in its optimization phase, leading to unexpected jumps while stepping through the code.

CHART

When attaching a debugger to a chart document, the chart will display a corresponding hint below the standard legend.



Debugged chart

EQUILLA PROGRAM STRUCTURE AND SYNTAX

META TAGS

The so-called meta properties define the global functions of your indicator, for example if the indicator should be displayed in a window of its own or within another chart. You can also add information about how the indicator is calculated (Synopsis) and define a ShortCode. The meta information area can usually be found in the beginning of the source code. You should adhere to this convention. The area has to be introduced with the tag

Meta:

The three most important pieces of meta information are:

Meta:

```
//Description for the code
Synopsis( " Text " ),
//Define whether the indicator will be displayed in a separate sub-chart or not
Subchart(True/False),
//Define a ShortCode. By entering it in the command linethe indicator is called up.
ShortCode("MyIndicator");
```

INPUT DECLARATION

When applying an indicator to a chart, one or more parameters are available in the indicator properties. These parameters can be calculation periods, multiplications or lists. User-specific parameters have to be declared in the input area of the source code. The area has to be introduced with the tag

Inputs:

AVAILABLE INPUT TYPES

Inputs:

```
//Switch for true/false /yes/no) choice.
Visuals (true, false),

//Input field for integers, followed by lower and upper
//end of range.
```

```

Period(10, 1, 100),

//Input field for decimal value; note that the
//decimal point of the defining criterion.
FactorOffset( 1.25, 1.0, 5.0 ),

//Input field for string; note that
//the quotation marks are the defining criterion.
UserName( "John Smith" ),

//Input field for colors; the dialog is system-dependent.
//The dialog generation also works for colors entered as
//RGB values ( redValue, greenValue, blueValue ).
//The values for redValue, greenValue und blueValue have to be between
//0 and 255.
PreferredColor( red ),

//Generates a selection list for time lines. The list automatically
//includes all time lines for the basic symbol. Usually, these are the
//Open, High, Low, Close and Volume data.
PriceValue( Close ),

//Generates a selection list for the given words or strings.
//The list is zero-based.
Weekdays( Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday );

```

EXAMPLES FOR USING THE INPUT TYPES

```

//Example for querying a true/false (yes/no) boolean switch
//When the switch is set to true (yes), a line is drawn:
If Visuals = True Then
    DrawLine ( (High+Low)/2 );

//Example for using an integer input
//The parameter 'Period' is declared as input, the given
//value 10 for the period is passed to the average function.
avgValue = Average( Close, Period );
Drawline( avgValue );

//Example for using a decimal number
//The drawing function draws a continuous line which
//corresponds to the value 'close * 1.25'.
DrawLine( FactorOffset * Close );

```

```
//Example for using a string input
//In the output window, 'John Smith' is displayed.
Print( "UserName" );

//Example for using a color value as input
//A red, solid line is drawn.
DrawLine( Close, "Close Line", StyleSolid, 2, PreferredColor );

//Example for using a time line input
//A line is drawn on the level of the close prices.
DrawLine( PriceValue );

//Example for querying the selected entry in a list
//In the output window, 'Monday' appears when selected.
If Weekdays = "Monday" Then
    Print( "Monday" );

//In lists, index numbers are assigned to the entries.
//As already mentioned, lists are zero-based, so Monday is "0" and Sunday is "6".
//Instead of querying the string, the index number can be used instead.
If Weekdays = 0 Then
    Print( "Monday" );
```

VARIABLE DECLARATION

WHAT ARE VARIABLES?

Variables are data containers. To be able to use values like numbers, calculation results, user entries or prices, the values have to be saved temporarily in variables. This way, each value has its own storage space and can be called up by the variable name. Imagine variable to be like containers on a cargo ship. Each container has its own unique ID, with which to find the contents. When the booker needs to know the information in a container, he enters the unique ID into a computer, which returns the wanted information.

VALIDITY OF VARIABLES

Variables are only valid within the script in a chart. It is not possible to use the value of variable "Period1" of the indicator "Indicator1" in "Indicator2", since the second script has no access to the variable - not even when both indicators are used in the same chart.

The only exceptions to the rule are global variables.

INITIALIZATION OF VARIABLES, AND FOR HOW LONG DO THEY KEEP THEIR VALUE?

Each Equilla script, e.g. an indicator, is processed once for each trading period. If the indicator is used in a daily chart, the indicator is calculated once for each day in the chart. The variable is initialized at the first trading day in the chart.

If in the days following that, no new value is set for the variable, the value is kept until the last trading day in the chart. If a new value is set for the variable in between, it is kept until a new value is assigned or until the last trading day.

GENERATING VARIABLES

Variables are generated by being declared, i.e. given a name. Variables can only be defined within a certain code area. It has to be introduced with the tag

```
Variables:
```

You have the following possible methods to declare a variable:

```
//Declare a variable by assigning a name
Variables:
    myFirstVariable, mySecondVariable;

//Declare a variable by assigning a name and value
Variables:
    myFirstVariable( 10 ), mySecondVariable( myFirstVariable / 2 );

//Declare a variable by entering a string
Variables:
    myFirstString ( "" ), mySecondString( "String test" );

//Declare a variable with color values
Variables:
    myFirstColor( red ), mySecondColor( RGB (10, 20, 30) );
```

GLOBAL (SCRIPT-INDEPENDENT) VARIABLES

As opposed to the statement above, you can use global variables under special circumstances. The limitation is that the scripts have to be in the same chart or portfolio. A sub chart, for example an indicator chart, does not count as a full chart. Therefore, you can pass data between main and sub charts and between several Equilla scripts that are used within the same main/sub charts.

To do so, global variables are available. A global variable consists of a "Namespace", which can be seen as the name of the cargo ship, and a variable name, which can be seen as the container ID. Since global variables are valid across several scripts, they do not have to be defined in the "Variables:" area. It is enough to name the global variable in one script and assign a value this way.

```
//This is the way to set up a global variable
```

```
//The global variable with the Namespace 'global' and the name
//'myFirstGlobalVar' is generated by assigning a value.
//From this moment on, the value can be used in all
//scripts in the same chart.
global::myFirstGlobalVar = 100;

//This is the way to read the value of a global variable.
//In the output window, '100' is displayed.
Variables:
    myNewVar;

myNewVar = global::myFirstGlobalVar ;
Print( myNewVar );
```

DECLARATION DETAIL FOR VARIABLES AND INPUTS

For variables as well as inputs, you can either enter several declarations in one line or each declaration in its own line. The declarations have to be separated by commas. After the last declaration, a semi-colon has to be entered.

```
//Declare inputs in one line
Inputs:
    MyName( "Caspar David" ), YourName( "Peter King" ), HerName ( "Barbara Myers" );

//Declare inputs in several lines
Inputs:
    MyName( "Caspar David" ),
    YourName( "Peter King" ),
    HerName ( "Barbara Myers" );
```

STRUCTURING SOURCE CODE

For a better readability and overview in the Equilla code, the source code should be well-structured. This also makes it easier to find errors in the coding. For structuring, make use of indentions, comments and empty lines. Declare the variables and inputs in separate lines instead of one line.

INDENTING SOURCE CODE

It is a standard programming technique to indent certain parts of the code by using the **Tab** key. This way it is much easier to understand the relations between code parts. This is especially useful in cases of loops or conditions.

```
//Indenting the code that is only run if the condition is met
```

```

If Bronto = "Saurus" Then
    DrawText( Close, "Run away!" );

//Cascading indenting of code to keep a better overview
If condition = "met" Then
    Begin
        myFirstVar = 100;
        mySecondVar = 7;
        DrawLine( myFirstVar + mySecondVar );
    End;

//Each consecutive condition should also be indented
If condition = "met" Then
    Begin
        myFirstVar = 100;
        mySecondVar = 7;
        //additional condition
        if Visuals = True Then
            DrawLine( myFirstVar + mySecondVar );
        End;
    End;

```

BRACKETING EXPRESSIONS

It is also recommended that you use bracketing for conditional queries or loops. This way, you avoid overly complicated code and malfunctioning which may result from logical errors.

```

//Expressions in queries should be bracketed for two reasons
//a) For better overview
If ( myFirstVar = 100 ) and ( mySecondVar = 100 ) Then
    Begin
        //Here follow...
        //...expressions
    End;

//b) For program logic
If (myFirstVar = 100) AND ((mySecondVar = 100) OR (myThirdVar = 0)) Then
    Begin
        //This block is only executed if
        //myFirstVar and mySecondVar both have the value 100
        //or if myFirstVar has the value 100 and if myThirdVar has the value 0
    End;

```

Take a closer look at the second example. If the two brackets around the expressions linked by OR were not there, another meaning would result. The condition would be met if "myFirstVar = 100 and mySecondVar = 100" OR "myThirdVar = 0". In this case, the code would only be executed if "myFirstVar" were 100 and "mySecondVar" were 100 OR "myThirdVar" were 0, which was not meant to be according to the definition given in the comments.

In addition to the examples above, it may also be useful to set brackets around mathematical operators.

```
//Are you sure what's meant here?
myFirstVar = 100 * ( x + y * ( 10 / z ) );

//This way it is easier to read.
myFirstVar = 100 * ( x + ( y * ( 10 / z ) ) );
```

INSERTING COMMENTS

As already seen, you can enter comments in the code. This is especially useful for two reasons

- Even in case of longer source code and more complex structures, good readability and comprehensibility are given. This way, you can correct or edit the code even after years.
- Outsiders are able to understand the code much faster.

```
//This is a single line comment

avgValue (Open) // You can also add a comment at the end of the line

/*This is the start of a multiple line comment Comments are disregarded by the program
interpreter, they are only displayed in the editor. With comments program parts can be
deactivated for example for error search. ---This assignment of a calculation to a parameter
is not executed Value1 = XAverage( Close, 10 ); This is the end of the comment */
```

EQUILLA FUNCTIONS

WHAT IS A FUNCTION?

In programming languages, functions are subordinate portions of the code in which regularly used calculations and subroutines are sourced out. For example, instead of writing the calculation of standard deviation again and again, the calculation is put into a function. By calling up the function by its name - and passing the relevant parameter(s) for calculation - the user receives the return value as result.

The call may look as follows:

```
/* With the following function you receive the standard deviation over a 10 day range.
'StandardDev' is the Equilla function that contains the actual calculation. */

myStdAbw = StandardDev( Close, 10 , 0 );
```

This is the source code of the Equilla *StandardDev*.

```
Meta:
    Synopsis( "[Standard Deviation 1] Returns the Standard Deviation value for the current
bar." );

Inputs:
    Price( NumericSeries ),
    Period( NumericSimple ),
    DataType( NumericSimple );

Variables:
    divisor, sumSquare, avg, i;

StandardDev = 0;

If DataType = 1 Then
    divisor = Period
Else
    divisor = Period - 1;

If divisor > 0 Then Begin
    avg = AverageFC( Price, Period );
    sumSquare = 0;
    For i = 0 To Period - 1 Do
        sumSquare = sumSquare + Sqr( Price[i] - avg );
```

```
If sumSquare > 0 Then
    StandardDev = Sqrt( sumSquare / divisor );
End
// *** Copyright Tradesignal GmbH ***
// *** www.tradesignal.com ***
```

Another big advantage of functions is that you can call up the same functions from indicators and strategies. In fact, all indicators are based on functions - just as strategies are based on indicators.

You can find the functions that came with the program by clicking on the button **Functions** in the toolbox. To see the code of the function, right-click on it to open the context and select **Edit**.

FUNCTION STRUCTURE

For the programming of Equilla functions, the same conditions apply as for all Equilla scripts, see the chapter Equilla Program Structure and Syntax. Minor divergences are given in the following paragraphs.

FUNCTION NAME

The function name is necessary to a) save the function in the file system and b) call up the function from within an Equilla script. When choosing the name, take care that the name

- is short and concise
- relates to the contents
- is not yet in use for another function (this might cause problems)

```
//The function Momentum gives the indicator of the same name as result
myFirstVar = Momentum( PriceSerie, CalculationPeriod );
```

PASSING PARAMETERS UPON FUNCTION CALL

For the calculation of indicators, statistics and other values, certain information has to be given:

- which data is to be used for calculation
- which time range is to be used for calculation

When calling up the function, these parameters (separated by commas) have to be passed in brackets.

```
/* The function 'Momentum' gives the indicator of the same name as result As parameter the
function demands the data to be used for the Momentum and the time range over which to
calculate. */

myFirstVar = Momentum( Close, 20 );
```

INPUT DECLARATION IN FUNCTIONS

In the source code of the function, any inputs from outside have to be declared. As opposed to the definition of indicators, the type of the input parameters has to be declared in functions.

Inputs:

```
//A data series, e.g. prices, has to be declared as NumericSeries
PriceValue( NumericSeries ),

//Simple numeric values are declared as NumericSimple
PeriodFast( NumericSimple ),

//Booleans as true/false (yes/no) value are declared TrueFalseSimple
DrawVisuals( TrueFalseSimple ),

//Strings like words or symbols are declared as StringSimple
UserName( StringSimple ),

//If you want to pass data that will be modified within the function and the
//resulting modified value is to be used outside of the function, you have
//to declare this as NumericRef
AdjustedPeriod( NumericRef ),

//Arrays can have multiple dimensions in Equilla, each dimension also
//declares an additional input for the function, in this example the
//function expects a 2-dimensional array as input which will be accessible
//with the name Array2D and the numeric inputs Rows and Columns which are
//initialized to contain the length of the array in this dimension.
Array2D [ Rows, Columns ] ( NumericArray ),

//If you want to pass an extension object to a function, you have to declare
//the input as ObjectSimple (available in Tradesignal 6.3 and above).
ExtensionObject ( ObjectSimple );
```

Make sure to use the parameters in the same sequence as they are declared in the function. The names of parameters used inside of functions are independent of parameter names that are passed. Therefore you can use the parameter 'Period', which is passed when calling the function, under another name within the function. You only have to take care that the sequence of values passed is the same as in the declaration.

VARIABLE DECLARATION IN FUNCTIONS

The same conventions apply as described in Equilla Basics. You can also use variable names that are already in use in the main

program, as there is no danger of collision.

RETURNING VALUES

Values that were calculated in the function usually should be available for further processing. To do so, the values have to be "returned". The actual way of returning depends on which kind of value has to be returned.

RETURNING A SINGLE VALUE

If only a single value has to be returned, this is usually done with the name of the function. In the calling script, the function is assigned to a declared variable. The variable takes up the calculation result and makes it available for further operations.

```
// In the calling script, the result of the Momentum calculation
// is assigned to the variable myFirstVar

myFirstVar = Momentum( Close, 20 );
```

In the source code of the function, the calculation result has to be assigned to the name of the function.

```
Meta:
    Synopsis( "[Momentum] Returns the momentum of a series over a specified period." );

Inputs:
    Price( NumericSeries ),
    Period( NumericSimple );

//Assignment, i.e. returning the value

Momentum = Price - Price[ Period ];
```

RETURNING SEVERAL VALUES

When you want to return more than one result from a function, you must use so called reference parameters. These parameters will be passed into your function, modified within the function and then can be used. The following function shows two such reference parameters that are set to the result of two different moving averages:

```
//Function: TwoSmoothMA
Inputs:
    FastMA(NumericRef), // Notice how we declare the type of the inputs as numeric
references
    SlowMA(NumericRef); // instead of numericseries or numericssimple.
```

Variables:

```
smoothMA;

smoothMA = Average(Close, 5);
FastMA = Average(smoothMA, 10);
SlowMA = Average(smoothMA, 50);
TwoSmoothMA = 1; // A dummy return value that will not be used in this case.
```

We can now use our new function in the following way:

```
//Indicator: Two Smooth Averages
Meta:
    SubChart(False);
Variables:
    slowMA, fastMA;

TwoSmoothMA(slowMA, fastMA); // The values for slowMA and fastMA are set in the function

DrawLine(slowMA, "Slow");
DrawLine(fastMA, "Fast");
```

The advantage of using multiple output functions is that related operations can be nicely encapsulated, but more importantly that if the result of each output rely on a shared calculated value, it is more efficient to do the shared part once rather than twice in two functions that return single values.

RETRIEVING PAST VALUES WITH DISPLACEMENTS

By using displacements, for example "Price[i]", you can retrieve older function results. A good example for this would be the search for a breakout within the last 20 days' high or low. Please note that with the displacement attribute, past values are retrieved. Therefore "[1]" gives the value for yesterday, [2] the value of two days ago etc.

Attention: Trying to retrieve values of functions that are not calculated continuously may result in wrong indicator results. This would be the case, for example, when the function call is in a subroutine that is not executed for each bar.

ADDITIONAL INFORMATION FOR FUNCTIONS

For all Equilla functions, you can find additional information in Tradesignal. To display it, open the source code in the Equilla Editor. Then right-click on the function in question, e.g. *DateTime*, and choose **Lookup Equilla Function** from the context menu. A

window opens with information on the command and links to related functions.

EQUILLA OPERATORS

An important part of the scripts is the calculation of values. Often only basic arithmetic is necessary for these calculations, but sometimes the tasks are more complex. Equilla places many operators for calculating and comparing values at your disposal.

CALCULATIONS

BASIC ARITHMETICS

For calculating values, the four basic arithmetic operations are available. These are used most often.

Variables:

```
myFirstVar, mySecondVar, myThirdVar;

//Adding two values
myFirstVar = mySecondVar + myThirdVar;

//Subtracting two values
myFirstVar = mySecondVar - myThirdVar;

//Multiplying two values
myFirstVar = mySecondVar * myThirdVar;

/* When dividing two values, care has to be taken that no division by zero occurs (which may
happen if price series are faulty and include a zero). In this case, Tradesignal instantly
reports an error and stops the calculation. After a simple example for a division, two
examples are given that are designed to prevent a division by zero. */

//Simple Division
myFirstVar = mySecondVar / myThirdVar;

//Query if Divisor = NULL
If myThirdVar <> 0 Then
    myFirstVar = mySecondVar / myThirdVar

//If myThirdVar is zero, the result of the
//preceding trading period is used.
//The alternative block starting with 'Else'
//may be left out
Else
    myFirstVar = myFirstVar [ 1 ];
```

```
//The Equilla function 'Divide' is available for this task
//The two operands have to be passed in brackets
myFirstVar = Divide( mySecondVar , myThirdVar );
```

MATHEMATICAL FUNCTIONS FOR COMPLEX CALCULATIONS

With some effort, every calculation can be performed. However, you can reduce the programming load by using functions already provided for special tasks. You can find a list of all functions by clicking the **Function** button in the toolbox. To see the code, double-click the function or select **Edit** from the context menu.

In the following, excerpts of some important functions and their use are given.

```
//No subchart
Meta:
    Subchart( false );

//Declare period for calculation and variables
Inputs:
    Period( 10 );

Variables:
    myFirstVar, mySecondVar, myThirdVar;

/* This way you can find the highest value in a series. Please note: Parameter 1: The used
data has to be given as a continuous series, either a price series or a value that is
calculated or existent at each bar. Parameter 2: The calculation span gives the number of
values from the past that are used in the calculation. In the example, period is set to
'10'. */

//Example 1: finding the highest close
myFirstVar = HHV( Close, Period );

//Example 2: finding the highest price span
mySecondVar = High - Low;
myFirstVar = HHV( mySecondVar, Period );
/* There is a similar function you can use for finding the lowest price in a series. Besides
the different function name, all statements from above apply. */

//Example 1: finding the lowest close price
myFirstVar = LLV( Close, Period );

//Example 2: finding the lowest price span
mySecondVar = High - Low;
```

```
myFirstVar = LLV( mySecondVar, Period );
```

```
/* When working with data, you often do not know during the programming which values are
less or greater than others. However, it is often important to pursue calculations with the
greater or lesser of two values. For this task, the two functions MaxItems and MinItems are
available. In both cases, two or more comma-separated values have to be passed in brackets.
The functions then give the greatest or smallest value from a list of values. */
```

```
//Calculating the height of a candlestick body
//without knowing if the close is below or above the open
```

```
myFirstVar = MaxItems( Open, Close ) - MinItems( Open, Close );
```

```
//Extracting the decimal places of a value
```

```
If isLastBar Then
    Begin
        myFirstVar = Frac( 5.5 );
        Print( myFirstVar ); //0,5 is output
    End;
```

```
//Finding the biggest integer that is smaller than the given value
```

```
If isLastBar Then
    Begin
        myFirstVar = Floor( 5.5 );
        Print( myFirstVar ); //5 is output
    End;
```

```
//Finding the smallest integer that is greater than the given value
```

```
If isLastBar Then
    Begin
        myFirstVar = Ceiling( 5.5 );
        Print( myFirstVar ); //6 is output
    End;
```

```
//Calculating the cosine of a value
```

```
If isLastBar Then
    Begin
        myFirstVar = Cos( 5.5 );
        Print( myFirstVar ); //0,7086697742913 is output
    End;
```

COMPARISONS

When calculating an indicator, it is often necessary to compare values and then execute certain functions. For value comparison, several operators are available.

SIMPLE COMPARISONS BETWEEN TWO VALUES

```
//No subchart
Meta:
    Subchart( false );

//Declare variables
Variables:
    myFirstVar, mySecondVar, myThirdVar;

//Query if two values are identical
If isLastBar Then
    Begin
        If myFirstVar = mySecondVar Then
            Print( "Hello World" ); //Data is output if values are identical
        End;
    End;

//Query if two values are unequal
If isLastBar Then
    Begin
        If myFirstVar <> mySecondVar Then
            Print( "Hello World" ); // Data is output if values are unequal
        End;
    End;

//Query if the first value is greater or equal to the second
If isLastBar Then
    Begin
        If myFirstVar >= mySecondVar Then
            Print( "Hello World" ); // Data is output if the value is
            // greater or equal to the
            // second value
        End;
    End;

//Query if the first value is smaller or equal to the second
If isLastBar Then
    Begin
        If myFirstVar <= mySecondVar Then
            Print( "Hello World" ); // Data is output if the value is
            // smaller or equal to the
            // second value
        End;
    End;
```

COMPARISONS BETWEEN SEVERAL VALUES THROUGH COMBINATION

```
//No subchart
Meta:
    Subchart( false );

//Declare variables
Variables:
    myFirstVar, mySecondVar, myThirdVar;

//Combining two comparisons with the AND operator
If isLastBar Then
    Begin
        If ( myFirstVar = mySecondVar ) AND ( myFirstVar = myThirdVar ) Then
            Print( "Hello World" ); //Data is output if both comparisons are true
        End;

//Combining two comparisons with the OR operator
If isLastBar Then
    Begin
        If ( myFirstVar = mySecondVar ) OR ( myFirstVar = myThirdVar ) Then
            Print( "Hello World" ); //Data is output if one comparison or both
                                   //are true
        End;

//Combining two comparisons with the XOR operator
If isLastBar Then
    Begin
        If ( myFirstVar = mySecondVar ) XOR ( myFirstVar = myThirdVar ) Then
            Print( "Hello World" ); //Data is output if one comparison is true
                                   //but not both
        End;
```

FINDING CROSSING TIME SERIES

One of the most important problems in indicators and strategies is the search for intersections between prices and averages, indicators and alert lines etc. Tradesignal offers several functions for this task.

```
//No subchart
Meta:
    Subchart( false );
```

```
//Set periods for calculation
Inputs:
    PeriodFast( 10 , 1 ),
    PeriodSlow( 20 , 1 );

//Declare variables
Variables:
    fastAvg, slowAvg;

//Calculating a short average of close prices
fastAvg = Average( Close, PeriodFast );
//Calculating a longer average of close prices
slowAvg = Average( Close, PeriodSlow );

If isLastbar then
    Begin
        //Query if the faster average crosses over slower
        If fastAvg Crosses Over slowAvg Then
            Alert( "Average moves upwards" );
        //Query if the faster average crosses under slower
        If fastAvg Crosses Under slowAvg Then
            Alert( "Average moves downwards" );
    End;
```

EQUILLA LOOPS

Loops serve to call functions multiple times without having to enter the function code again and again. This is especially useful in cases in which the number of necessary runs for the loop is calculated during program execution.

In this chapter, all loop types available in Equilla are described.

For all Equilla operators, you can find additional information in Tradesignal. To display it, open the source code in the Equilla Editor. Then right-click on the operator in question, e.g. *Begin*, and choose **Lookup Equilla Function** from the context menu. A window opens with information on the operators and links to related functions.

LOOPS WITH A DEFINITE NUMBER OF CYCLES (FOR...TO...DO)

For this type of loop, start and end values are given. The number of cycles is given by their difference.

```
//This loop repeats the command 10 times
For counter = 0 to 9 Do
    command;

//This loop repeats the block of commands 10 times
For counter = 0 to 9 Do
    Begin
        command1;
        command2;
        ...
        commandX;
    End;
```

You also have the option to rule with which steps the loop have to run. The standard for the step is 1.

```
//This loop repeats the command 5 times
For counter = 0 to 9 Step 2 Do
    command;

//This loop repeats the command 10 times with a negative incrementation
For counter = 9 to 0 Step -1 Do
    Print(Counter);
```

You also have the option to give the start and end value of the loop as variables. This way, the loop is more flexible and results of partial calculations or functions can be used as counters.

```
//This loop repeats the command 13 times
Variables:
    startCount( 0 ), endCount( 12 );

For counter = startCount to endCount Do
    command;

//This loop repeats the block of commands 13 times
For counter = startCount to endCount Do
    Begin
        command1;
        command2;
        ...
        commandX;
    End;
```

CONDITION-CONTROLLED LOOPS

The above presented type of loop uses a fixed value as a counter for the number of cycles. The values can be set by variables, but these variables cannot be changed from within the loop. However, this is necessary in cases, for example, in which the number of cycles depends on the result of a calculation within the loop itself.

For these cases, other types of loops exist. These loops run until a certain result or value in the cycle meets a certain condition. There are two different subtypes of these loops:

LOOP WITH THE CONDITION BEFORE THE EXECUTION (WHILE...DO)

The condition for the loop is checked before the loop is executed.

```
//This loop runs 10 times ( 0-9 )
Variables:
    counter(10), sumCounter(0);

While ( counter < sumCounter ) Do
    sumCounter = sumCounter + 1;
```

```
//You can also run several commands as a block.
```

Variables:

```
counter(10), sumCounter(0);
```

```
While ( counter < sumCounter ) Do
```

```
Begin
```

```
    sumCounter = sumCounter + 1;
```

```
    Print( sumCounter );
```

```
End;
```

LOOPS WITH THE CONDITION AFTER THE EXECUTION (REPEAT...UNTIL)

The condition for the loop is checked after each cycle run, i.e. after the calculation.

```
//This loop runs 10 times ( 0-9 )
```

Variables:

```
counter(10), sumCounter(0);
```

```
Repeat
```

```
    sumCounter = sumCounter + 1;
```

```
Until ( sumCounter < counter );
```

```
//You can also run several commands as a block.
```

Variables:

```
counter(10), sumCounter(0);
```

```
Repeat
```

```
Begin
```

```
    sumCounter = sumCounter + 1;
```

```
    Print( sumCounter );
```

```
End;
```

```
Until ( sumCounter < counter );
```

EQUILLA CONDITIONS

One of the most important aspects of programming indicators, strategies or other routines is the reaction to certain events, like calculation results or user inputs. Without the control, activation or deactivation of program parts depending on those events, no complex programming would be possible.

For all Equilla conditions, you can find additional information in TradeSignal. To display it, open the source code in the Equilla Editor. Then right-click on the operator in question, e.g. *Else*, and choose **Lookup Equilla Function** from the context menu. A window opens with information on the operators and links to related functions.

SIMPLE CONDITION (IF ... THEN)

You can use the key words **If...Then** to encapsulate one or more conditions. Depending on the results, different actions are performed. The conditions can be combined by using the three operators And, Or and Xor (see the chapter Equilla Operators).

```
//No subchart
Meta:
    Subchart( false );

//A list with two input parameters is declared

Inputs:
    commodities( Silver, Gold );

//Query if the first list entry is selected
If Commodities = 0 Then
    Print( "Silver price rises." )
Else //If not selected, the alternative is processed
    Print( "Gold price rises." );
```

CONDITIONS WITH BRANCHING (IF...THEN...ELSE)

By entering an alternative command, your script can react to met and not met conditions likewise. When testing several cases at once, it is sensible to process them within the same context.

```
//No subchart
Meta:
    Subchart( false );

//A list with three input parameters is declared
```

Inputs:

```

    commodities( Silver, Gold, Platin );

//Query if the first list entry is selected
If Commodities = 0 Then
    Print( "Silver price rises." )
Else //If not selected, the alternative is processed
    Print( "Gold price rises." );

//Query if the first list entry is selected
If Commodities = 0 Then
    Print( "Silver price rises." )
Else If Commodities = 1 Then //If not selected, the second alternative is checked
    Print( "Gold price rises." )
Else //Is it not selected, only the last possibility is left.
    Print( "Platin price rises." );

```

Please note that the line before the Else statement must not end with a semi-colon.

NESTED CONDITIONS

You can put several statements into a block and nest queries. The following example shows how to write begin-end blocks and nested conditions.

```

//Important, this is not a working program. Do not attempt
//to compile it in the editor.
//We check the cucumber and want to buy a really fresh one only
If cucumber = green Then
    Begin
        //The color meets our first condition, but is it also fresh?
        If ( cucumber = fresh ) Then
            Begin
                //The cucumber is fresh, we use it
                "Cut in slices";
                "Add tomato slices";
                "Add dressing";
                //If it's Friday, add garlic
                If Today = Friday Then
                    "Put lots of garlic in the dressing"
                Else //If it's not Friday, do not use garlic
                    "Skip garlic";
            End
        End
    End

```

```
Else //The cucumber was too soft and squashy, we try to buy another one.  
    "Drive to another store";  
End;
```

EQUILLA QUOTATIONS AND TIME PERIODS

This entry is about quotations that are delivered by the provider and that you need as input values for the calculation of functions, indicators and strategies. It also offers information on working with date and time date. In the end, you can find information and examples on how to find distinct bars in a chart.

For all Equilla functions, you can find additional information in Tradesignal. To call it up, open the source code in the Equilla Editor. Then right-click on the function in question, e.g. *Accelerator*, and choose **Lookup Equilla Function** from the context menu. A window opens with information on the function and links to related functions.

DATA IN QUOTATION SERIES

The most important data needed for indicators are the quotations. These are delivered by the data provider and can be displayed in a chart. In addition, you can use external quotations from locally saved files in Tradesignal. From most providers, you will receive the following data:

- **Open** - Opening price for each trading period
- **High** - High price for each trading period
- **Low** - Low price for each trading period
- **Close** - Closing price for each trading period
- **Volume** - Trading volume for each trading period
- **Open Interest** - Open interest in the case of futures

PARTICULARITIES OF VOLUME DATA

Two different definitions of trading volume are in existence, which are often confused.

- The more meaningful definition is that the trading volume is given by the *number of traded stocks multiplied with its traded price*. This way, the volume represents the amount of capital that was traded in the period.
- Differing from this, sometimes the trading volume is only given as *turnover in traded pieces*. This means only the number of traded stocks is taken as volume, without information on the traded prices. The problem is that in this case, 100 stocks at 0.20 cents would yield the same volume as 100 stocks traded at \$164.40. The number of pieces does not represent the capital used for the trades.

Make sure to ask your data provider which kind of volume data is delivered. Depending on the volume interpretation, indicator data and statistics have to be adapted or put into perspective.

EXAMPLE: USING QUOTATIONS SERIES WITHOUT VOLUME

```
/* The following example demonstrates the usage of quotations series without volume. Four variables are declared which will be used for each of the quotation series. Following this, an average is calculated for each series, saved in the variables and displayed as a candlestick chart in a new window. Remember that Equilla scripts are processed once for each
```

```
trading period. */

Meta:
    Subchart ( true );

//The trading period for the average calculation is declared as input.
//You can later edit the value in the properties of the indicator.

Inputs:
    Period( 10, 1 );

//Variables are declared

Variables:
    myOpenValue, myCloseValue, myLowValue, myHighValue,
    avgOpen, avgClose, avgLow, avgHigh;

//Day for day is processed. For each trading day, the price is assigned to
//the variables.
myOpenValue = Open;
myCloseValue = Close;
myLowValue = Low;
myHighValue = High;

/* Now the average calculation follows, which is done once per trading day. The passing of
the quotations to the calculation functions is especially complicated to demonstrate the
underlying principle. Alternatively, you can pass the quotations to the functions directly.
avgOpen = Average ( Open, Period ); avgClose = Average ( Close, Period ); avgLow = Average
( Low, Period ); avgHigh = Average ( High, Period ); */

avgOpen = Average ( myOpenValue , Period );
avgClose = Average ( myCloseValue, Period );
avgLow = Average ( myLowValue, Period );
avgHigh = Average ( myHighValue, Period );

//With the result of the average calculations
//candlesticks are drawn
DrawCandleStick( avgOpen, avgHigh, avgLow, avgClose, Yellow, Red, DarkBlue );
```

INLINE INSTRUMENTS AND DATA INPUTS

Indicators can be used for more than just the quotations of a single basic symbol in the chart. Many indicators use the quotation series of more than one stock. The same applies to strategies, many of which are based on several symbols or different quotation series. For these calculations, the tools Inline Instruments and Data Inputs are available.

INLINE INSTRUMENTS

An inline instrument is a reference to the stock data which is used directly in the source code. You have to give the symbol and the relevant quotation series. The following example calculates the difference between the quotations of Puma and Adidas Salomon. The result is displayed in a candlestick chart. It does not matter to which chart you apply this indicator - it always uses the given quotations of those two companies. Therefore, inline instruments can be used for calculation with quotations that are not a part of the current chart or portfolio.



Using Inline Instruments

```
//Generate a subchart
Meta:
    Subchart( true );

//Declare variables
Variables:
    openValue, highValue, lowValue, closeValue;

/* Subtract the time series. By giving the symbol in the series, quotations can be used that
are not part of the current chart. */
openValue = Open of 'pum ger' - Open of 'ads ger';
highValue = High of 'pum ger' - High of 'ads ger';
lowValue = Low of 'pum ger' - Low of 'ads ger';
closeValue = Close of 'pum ger' - Close of 'ads ger';

//Display result as candlesticks
DrawCandleStick( openValue, highValue, lowValue, closeValue, White, Red, Black );
```

DYNAMIC INLINE INSTRUMENTS

Tradesignal 6 offers you the ability to create inline instruments based on runtime information, e.g. the string inputs of an indicator or strategy may be used instead of a constant string expression.

See the **Equilla Formula Language** reference entry **Instruments** in Tradesignal for more details on importing symbols.

```
//Import a single instrument and reference it in the script by using DrawLine on its
variable
Input:
    Symbol( "" );

Instruments:
```

```
mySymbol( Symbol );

DrawLine( mySymbol );
```

You can also create a list of inline instruments and reference any of the symbols in the list via its index which starts at zero.

```
//Import a list of instruments and calculate the average price
Input:
    ListSymbol( "" );

Instruments:
    myInstruments( List( ListSymbol ) );

Variables:
    prices( 0 ), i( 0 );

prices = 0;

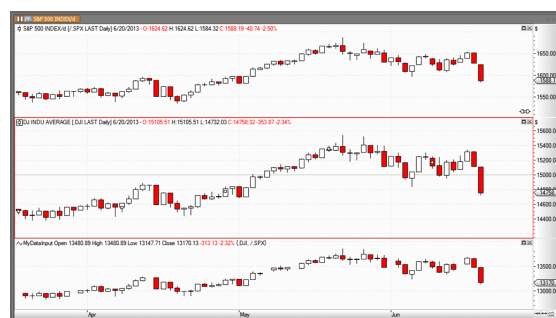
For i = 0 to Len( myInstruments ) - 1 Begin
    prices = prices + Close of myInstruments( i );
End;

If Len( myInstruments ) != 0 Then
    DrawLine( prices / Len( myInstruments ), "Average" );
```

DATA INPUTS

With data inputs, you can access different quotation series or other data in the chart or portfolio. The only condition is that those series are available in the same chart or portfolio during runtime. For example, you can write an indicator displaying the spread between two time series without defining beforehand to which symbols the series belong. You simply start a chart with two symbols and then apply the indicator.

To recreate the indicator from above with data inputs instead of input instruments, create a chart with Puma AG and Adidas Salomon AG. Now enter a new indicator. The quotations are used as data inputs. The first quotation series is referred to as "Data1", the second as "Data2" etc. Since up to twelve symbols can be displayed in a chart, you can enter up to 12 data inputs.



Using Data Inputs in a Chart

```
//Generate subchart
Meta:
```

```
Subchart (true);

//Using the data input parameters for the spread
DrawCandleStick( Open of Data1 - Open of Data2,
High of Data1 - High of Data2,
Low of Data1 - Low of Data2,
Close of Data1 - Close of Data2,
white, red, black);
```

DATE AND TIME FOR QUOTATION SERIES

The information is delivered with the price data, online or in files. For correct display of quotations, date and time are necessary. This information is also often necessary for other functions, e.g. to recognize day change-overs or certain days of time.

HOW A CHART IS DRAWN

The quotations are displayed linearly, from the first period (in the past) to the last. Usually, this is the current day or hour, depending on the time base of the chart. Within an Equilla script, the navigation is done with the consecutive number of the period (not the date or time). Therefore, if you want to use the close price of three days ago in your indicator, you need to write the script so that it goes three periods back in time.

The date cannot be used for this kind of navigation. You cannot retrieve data such as price or volume with a date. However, you can read the date and time of each period for other uses, for example the output of date and time of certain price levels in a chart.

WHICH DATA IS OFFERED BY TRADESIGNAL?

- **Date** - The date for the current period as numeric value: 1060609 (this is the 09.06.2006)
- **Time** - The time of the current period as numeric value: 1645 (this is 16:45)
- **DateTime** - A combined expression that is already formatted: 2006/07/05 17:25:00.000 (this is the 05.07.2006 1725)

Extracting the data with code:

```
//Date of the current trading day in a daily chart
Variables:
    myDate;

myDate = Date

//In the output window, the date for each trading day in the chart
//is displayed continuously.
Print( myDate );
```

```
//Time of the current period in an intraday chart
Variables:
    myTime;

myTime= Time;

//In the output window, the time for each trading period in the chart
//is displayed continuously.
Print( myTime);
```

FUNCTIONS FOR FORMATTING AND WORKING WITH DATE AND TIME

Above, the three basic time series for date and time are given. These data can be read for every period and be used further. In the example above, the date and time were read and printed in the output window. However, it is often necessary to find out single components of the date, for example the day of week, the calendar month or the current year. You may also want to use the given date and time values in another format than the one offered by the system.

FUNCTIONS FOR SPLITTING DATE AND TIME

The following functions are not a complete list of all available functions. Only the most important ones are listed.

For returning date components, you can use the simple functions *Year(date)*, *Month(date)* etc. When entering "now" for the date, the current date is used.

Instead of *day*, the function *DayOfMonth* can be used.

```
//Example: Write the components of the current date and time to the output window:
If IsLastBar Then
    Begin
        ClearDebug();
        Print( "Year=" + CStr( Year( Now ) ) );
        Print( "Month=" + CStr( Month( Now ) ) );
        Print( "Day=" + CStr( Day( Now ) ) );
        Print( "Hour=" + CStr( Hour( Now ) ) );
        Print( "Minute=" + CStr( Minute( Now ) ) );
        Print( "Second=" + CStr( Second( Now ) ) );
        Print( "MilliSecond=" + CStr( MilliSecond( Now ) ) );
    End;
```

To return parts of the *DateTime*, the function *DatePart* has to be used, defined as *DatePart(DateInterval, DateTime)*. For the definition of *DateTime*, see above.

The *DateInterval* is a numeric constant representing a certain part of the DateTime. It is always constructed with a leading *Interval* followed by a date or time expression.

Interval	Time
Interval_Year	Year
Interval_Month	Month of Year (1-12)
Interval_Day	Day of Month (1-31)
Interval_DayOfMonth	Day of Month (1-31)
Interval_DayOfWeek	Day of Week (0-6)
Interval_DayOfYear	Day of Year (1-365)
Interval_Hour	Hours
Interval_Minute	Minutes
Interval_Seconds	Seconds

```
//Example: Write a date or time part to the output window:
```

Input:

```
Interval( iYear, iMonth, iDay, iHour, iMinute, iSecond );
```

Variables:

```
name, part;
```

```
If Interval = iYear Then Begin
    name = "Year";
    part = Interval_Year;
End Else If Interval = iMonth Then Begin
    name = "Month";
    part = Interval_Month;
End Else If Interval = iDay Then Begin
    name = "Day";
    part = Interval_Day;
End Else If Interval = iHour Then Begin
    name = "Hour";
    part = Interval_Hour;
End Else If Interval = iMinute Then Begin
    name = "Minute";
    part = Interval_Minute;
End Else If Interval = iSecond Then Begin
    name = "Second";
    part = Interval_Second;
End;
```

```
Print( name + "=" + CStr( DatePart( part, DateTime ) ) );
```

FUNCTIONS FOR FORMATTING DATE AND TIME DATA

The basic date and time entries are dry figures. The DateTime component is also not formatted according to European standard. To allow you to generate various date and time formats, complex formatting functions are available. Two important ones are described here.

```
/* With the following function, you can format a date
so that it includes the full name of the month. The
function has the DateTime parameter as input. The
formatting possibilities are quite unlimited. */
```

```
//No subchart; output in the main chart
```

Meta:

```
Subchart ( false );
```

```
//Declaring variables for date and day
```

Variables:

```
myDate, myDay;
```

```
//Reformatting the date, for example to
```

```
myDate = FormatDate ( DateTime, "dd/mm/yyyy");
```

```
//Here the day is read
```

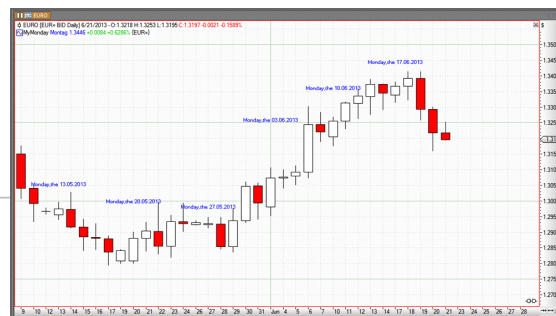
```
myDay = DayOfWeek ( Date );
```

```
//Check if it is Monday - if yes, the text is
```

```
//output in the chart
```

```
if myDay = 1 Then
```

```
DrawText ( High + Range(), "Monday", "Monday," + mydate, 10, blue );
```



Recognize Monday



Display the time in a chart

```
/* The following function is similar to the FormatDate function. It serves for formatting
the time data in user-defined formats. It is recommended to take a look at the Equilla help
for further information. */
```

```
//No subchart; output in the main chart
```

Meta:

```
Subchart( false );
```

```
//Declaring the variables for time and the formatted time
Variables:
    myTime, myFormatTime;

//Here the time is read
myTime = Time;

//Here the time is reformatted with a separator
myFormatTime = FormatTime( DateTime, "hh:mm");

//At five past nine, a message is displayed in the chart
If Time = 0905 Then
    DrawText ( High + Range(), "Good morning!", "It is " +myFormatTime,
10, blue);

//At one o'clock, it's time for the lunch break
If Time = 1300 Then
    DrawText ( High + Range(), "Lunch time!", "Go and have something to eat, it is "
+myFormatTime, 10, blue);

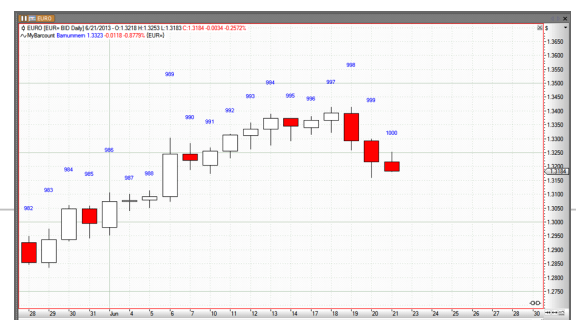
//At half past five, it's time for knocking off
If Time = 1730 Then
    DrawText ( High + Range(), "Bye!", "Call it a day, it is " +myFormatTime,
10, blue);
```

INFORMATION ABOUT SPECIFIC PERIODS AND BARS

In Equilla Graphs, Outputs and Alerts you can find information on how the navigation in a chart can be controlled by an Equilla script. Reference values are used that can also be interpreted as step sizes. Much as in a game, where a figure makes 6 steps forwards or backwards, the program navigates through the time series.

```
//Reading the date of three days ago
myPrevDate = Date[ 3 ];

//Reading the close price of five days ago
myPrevClose = Close[ 5 ];
```



Displaying continuous bar numbers

From each period in a chart, you have access to all other data and prices in the chart by going back or forward in the data. To do so, some useful functions are available in Tradesignal, which can be used with Equilla.

```

/* The parameter 'CurrentBar' gives the reference number of the current period The number is
listed consecutively, as can be seen in the chart. */

//Output to be done in the main chart
Meta:
    Subchart( false );

Variables:
    myBarNum;

//The numeric value CurrentBar is converted into a string
myBarNum = CStr( CurrentBar );

DrawText( High + Range(), "Bar numbers", myBarNum, 10, blue);

```

For many applications it is important to execute a specific action at the very beginning or ending of a chart. For example, variables are often set at the first bar. Two important functions are offered by Tradesignal for this purpose.

- With the first function, you can read out how many bars (periods, candlesticks) are loaded in the chart.
- With the second function, a flag is set which is "true" when the last bar is reached.

```

//We set two variables at the first bar
Variables:
    myEntryCondition, myExitCondition, myBarNums;

//We look for the first bar
If CurrentBar = 1 Then
    Begin
        myEntryCondition = false;
        myExitCondition = false;
    End;

/* We look for the last bar. This time we use the number of total bars in the chart which we
get with the variable 'LastBar'. */

If CurrentBar = LastBar Then
    Begin
        myBarNums = LastBar;
        Print ( myBarNums );
    End;

//To find the last bar, you can also use the
//function 'IsLastBar' which is a boolean.

```

```
If isLastBar = True Then
    Begin
        myBarNums = LastBar;
        Print ( myBarNums );
    End;
```

Last not least it is often important to know the current time base of the indicator. For this, the constant "TimeBase" is available. With it you get the number of seconds spanned by each bar. With some mathematics, we can quickly find out if the indicator is in a day chart or one with a five minute span.

```
Variables:
    outputString("");

If CurrentBar = 1 Then
    Begin
        //Checking for a day chart
        if TimeBase = 86400 Then
            outputString="d";
        //Checking for a one hour time base
        if TimeBase = 3600 Then
            outputString="1h";
        //Checking for a five minute time base
        if TimeBase = 300 Then
            outputString="5m";
        //Checking for a one minute time base
        if TimeBase = 60 Then
            outputString="1m";
    End;

//At the last bar, the output is written
//into the output window
If isLastBar Then
    Print ( outputString );
```

EQUILLA GRAPHS, OUTPUTS AND ALERTS

There are several ways to offer data, calculation results, symbols and reports to the users. One of the most important functions is the graphic display of indicators and other information. Tradesignal offers output functions for text and reports and an alert function.

For all Equilla operators, you can find additional information in Tradesignal. To display it, open the source code in the Equilla Editor. Then right-click on the operator in question, e.g. *Begin*, and choose **Lookup Equilla Function** from the context menu. A window opens with information on the operators and links to related functions.

DRAWING LINES, AREAS, SYMBOLS OR TEXT IN A CHART

With the drawing statements, usually indicators and graphs related to them are generated. An indicator usually consists of a line or histogram. Some indicators also use filled areas or symbols or offer information as text.

In this chapter, you can find general information on using colors, line width, symbols and their most important applications. When you enter a drawing statement in the editor, you can find all parameters and their sequence in the tooltip. Parameters that are not mandatory are labeled with "optional".

BASICS OF DRAWING STATEMENTS

Here you can find information on using drawing statements, why they should have captions and how the displacement function can be used to move the graphic display in the timeline. The most important parameters are given.

CAPTIONS

A caption should be entered for each statement. This is especially useful when several drawing statements are used in the same chart. The captions have two important functions:

- The drawing statements are more easily identified in the property inspector.
- In the tooltips and the target cursor mode in the chart, values are displayed for all graphic objects under the cursor. The value interpretation is easier when the drawing statements are named.

If no caption is given, each drawing statement is labeled with "Plot" and a consecutive number.

DISPLACEMENT

Equilla scripts are executed once per bar. By using so-called displacements, for example "Price[i]", you can retrieve older function results. In a similar way, drawing statements can be displaced. For this, the Displacement parameter has to be added directly after the statement. Enter a positive value in square brackets [] to move it into the past, and a negative value to move it into the future.

Example for a displacement:



Displacement example

```
// Generate no subchart
Meta:
    Subchart( false );

//Execute at last bar
If isLastBar Then
    Begin
        DrawSymbol[3] (Low[3], "Past", SymbolSquare, 30, black, blue );
        DrawSymbol[-3] (Low, "Future", SymbolSquare, 30, black, blue );
    End;
```

PARAMETERS

For all drawing statements, optional parameters can be specified. These are caption, line width, symbol size, colors etc. These parameters can be given by variables which can be changed at runtime (see some of the examples below). Feel free to experiment with the options.

Especially interesting is using dynamic color values. Each component of the color values of RGB and HSV modes can be changed. The opacity can also be set.

Design parameters like line or symbols styles can also be entered numerically. However, while this saves space, the full parameter names are easier to read and remember.

DRAWING STATEMENTS AND SYNTAX

Below, the most commonly used drawing statements and their syntax are listed. In the thumbnails, screenshots with graphs and source code are given.

DRAWLINE ()

With this statement, a line is drawn. Style, size and color can be specified. Constant values or definable parameters (e.g. results or calculations) can be used.

```
DrawLine[ Displacement ]( Value, "caption", Style,
LineWidth, Color );
```



Line example

Example for a sine curve:

```
// Generate subchart
Meta:
    Subchart( true );

//Declare variables
Variables:
    mySinValue, counter;

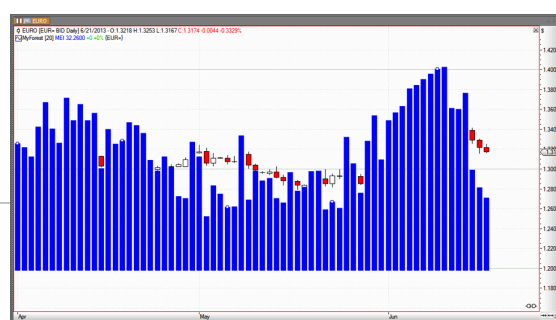
//Assign a value to the variable
mySinValue = Sin(counter);
counter = counter + 1;

//Draw the values as a solid red 2 pixel line
DrawLine( mySinValue, "Sinus", StyleSolid,2, Red );
```

DRAWFOREST ()

With this statement, a forest graph (histogram) is drawn. Typically the start value is the base line. The color, start and end point of needles can be specified. This way, not only can indicators be drawn but the forest can also be used for generating grids.

```
DrawForest[ Displacement ]( StartValue, EndValue,
"caption1", "caption2",
NeedleSize, NeedleColor, LinkForest True/False );
```



Forest example

Example for an indicator displayed as forest:

```
// Generate no subchart
Meta:
    Subchart( false );
```

Inputs:

```
Period( 20, 1 );
```

```
//Declare variables
```

Variables:

```
myOwnIndicator;
```

```
//Create the indicator
```

```
myOwnIndicator = RSI ( Momentum (Close, Period), 5 );
```

```
//Display indicator as forest
```

```
DrawForest( 0, myOwnIndicator, "Basis", "MEI", thick, blue, false );
```

DRAWAREA ()

With this statement, an area is drawn between two values of the scale.
Colors and opacity can be set.

```
DrawArea[ Displacement ]( Value1, Value2, "caption1",  
"caption2", FillColor,  
LineColor1, LineColor2 );
```



Area example

Example for drawing an area with color gradient:

```
// Generate a subchart
```

Meta:

```
Subchart( true );
```

```
//Declare variables
```

Variables:

```
counter, colorValue;
```

```
//Change the counter in a loop
```

```
if counter <= 254 then
```

```
    counter = counter + 1;
```

```
//Reset the counter at 255
```

```
Else
```

```
    counter = 0;
```

```
//Set the RGB color value with a dynamic value.
```

```
colorValue = RGB( counter, 150, 150 );

//Set opacity.

colorValue = TransparentColor ( colorValue, counter );

//Draw the area between the positive and negative values of
//'counter', with black border lines.

DrawArea( counter, Neg( counter ), "Up", "Down", colorValue, Black, Black);
```

DRAWSYMBOL ()

With this statement, symbol charts with circles, start, dots, triangles etc. can be drawn. The size and colors can be controlled dynamically.

```
DrawSymbol[ Displacement ]( Value, "caption",
SymbolStyle, SymbolSize,
SymbolBorderColor, SymbolFillColor );
```



Symbols example

Example for drawing triangles (pointing up or down, depending on the curves):

```
// Generate no subchart
Meta:
    Subchart( false );

//Declare input
Inputs:
    Period( 20, 1 );

//Declare variables
Variables:
    avgValue;

//Calculate the average close price
avgValue = XAverage ( Close, Period );

//Display an triangle pointing upward, if
//the price crosses above the average.
If Close Crosses over avgValue Then
    DrawSymbol( High, "CrossUp", SymbolTriangleUp, 20, black, darkgreen);
```

```
//Display an triangle pointing downward, if
//the price crosses below the average.
If Close Crosses under avgValue Then
    DrawSymbol( Low, "CrossUp", SymbolTriangleDown, 20, black, darkgreen);

//The average itself is displayed as line.
DrawLine( avgValue, "Average", StyleSolid, 2, black );
```

DRAWTEXT ()

With this statement, text is output. Variables like font size, color, style etc. can be specified.

```
DrawText[ Displacement ]( Value, "caption",
textoutput, FontSize, FontColor,
FontStyle, URL );
```



Output text example

Example for text output:

```
//Generate no subchart
Meta:
    Subchart( false );

//Declare input
Inputs:
    Period( 10, 1 );

//Declare variables
Variables:
    avgValue;

//Calculate the average close price
avgValue = WAverage ( Close, Period );

//When the average turns up, text is added to the
//Low of the last day.
If TurnsUp( avgValue, 2, 1 ) Then
    DrawText[1]( Low[1], "Up", "Bottom", 10, Darkgreen);

//When the average turns down, text is added to the
//High of the last day.
If TurnsDown( avgValue, 2, 1 ) Then
    DrawText[1]( High[1], "Down", "Top", 10, Red );
```

DRAWBAR ()

With this statement, a bar chart is drawn. Each of the four values - open, high, low, close - can be chosen freely. This way, indicators can also be drawn as bars.

In this statement no caption needs to be specified, as it can only be used once in each script.

```
DrawBar[ Displacement ]( Open, High, Low, Close,
BullishColor, BearishColor )
```



Bar chart example

Example for modified OHLC values as bars (Heikin Ashi method):

```
//Generate a subchart
Meta:
    Subchart( true );

//Declare input
Inputs:
    Period( 14, 1 );

//Declare variables
Variables:
    avgOpen, avgHigh, avgLow, avgClose;

//Calculate the average for all quotes.
avgOpen = XAverage( Open, Period );
avgHigh = XAverage( High, Period );
avgLow = XAverage( Low, Period );
avgClose = XAverage( Close, Period );

//Display bars.
DrawBar ( avgOpen, avgHigh, avgLow, avgClose, DarkGreen, Red );
```

DRAWCANDLESTICK ()

With this statement, a candlestick chart is drawn. Each of the four values - open, high, low, close - can be set freely. This way, indicators can also be drawn as candlesticks.

In this statement no caption needs to be specified, as it can only be used once in each script.

```
DrawCandleStick[ Displacement ] ( Open, High, Low,
Close, BullishColor, BearishColor,
BorderColor )
```



Candlesticks example

Example for modified OHLC display as candlesticks (Heikin Ashi method):

```
//Generate a subchart
Meta:
    Subchart( true );

//Declare input
Inputs:
    Period( 14, 1 );

//Declare variables
Variables:
    haOpen, haHigh, haLow, haClose;

//Calculate the Heikin Ashi values
If ( CurrentBar = 1 ) Then
    Begin
        haOpen = Open;
        haClose = Close;
    End
Else
    Begin
        haOpen = (haOpen[1] + haClose[1]) / 2;
        haClose = (open + high + low + close) / 4;
    End;

haHigh = MaxItems( high, haOpen, haClose );
haLow = MinItems( low, haOpen, haClose );

//Draw the Heikin Ashi values as candlesticks.
DrawCandlestick ( haOpen, haHigh, haLow, haClose, DarkGreen, Red, DarkGray );
```

DRAWPRICEMARKER ()

This draws a horizontal marker at the given price next to the price scale or aligned to the last bar. Price markers will only be drawn on the last bar of a chart. Additionally, price markers are only valid for one evaluation run. When an update occurs, all existing price markers will be deleted. Price markers are always drawn behind any other chart.

```
DrawPriceMarker( Price1, Price2, Width, FillColor, BorderColor, BorderThickness, Alignment,
Label, LabelSize, LabelColor, LabelOptions )
```

In contrast to drawing functions like DrawLine(), DrawPriceMarker () does not procure series data. Therefore the output from this function cannot be used from client scripts. Example for markers for all stop and limit order active on the current bar:

```
Meta:
SubChart( False );

Variables:
order, price, type, extent( 10 );

If IsLastBar Then Begin
For order = 1 To GetActiveOrderCount Begin
price = GetActiveOrderPrice( order );
type = GetActiveOrderType( order );
If type = OrderTypeLimit Then
DrawPriceMarker( price, price, extent, Blue, Blue )
Else If type = OrderTypeStop Then
DrawPriceMarker( price, price, extent, Red, Red );
End;
End;
```

DRAWING TOOLS WITH SPECIAL FUNCTIONS

For the two drawing tools **DrawTrendLine()** and **DrawRectangle()**, special functions are available in Equilla to draw with and manipulate these tools.

These special functions are passed as *(tool) flags*. They allow actions, for example expanding a rectangle into the "past" with "ToolLeftExtend".

In contrast to drawing functions like Drawline(), the output generated by the tool functions is valid for display in a chart only.

You can find more information on DrawTrendLine(), DrawRectangle() and the flags in Tradesignal help under **Equilla Formula Language**, menu entry **Tools**.

DRAWTRENDLINE ()

This draws a trendline into a chart.

```
DrawTrendline( StartDate, StartTime, StartPrice, EndDate, EndTime, EndPrice, Style, Width,
Color, Flags )
```

The flags are special functions like ToolExtendLeft, ToolExtendTop etc.

Example for a trendline drawn at the highest high of the last bars:

```
Input:
Price( High ),
Period( 10, 1 );

Variables:
highVal;

If IsLastBar Then
Begin
// Remove any trendline from the previous bars
While ( ToolGetFirst() <> -2 )
ToolDelete( ToolGetFirst() );

// calculate the highest high
highVal = HHV( Price, Period );

// draw the trendline
DrawTrendline( DateTime[10], highVal, DateTime, highVal );
End;
```

DRAWRECTANGLE ()

This draws a rectangle into the chart.

```
DrawRectangle( StartDate, StartTime, StartPrice, EndDate, EndTime, EndPrice, BorderColor,
FillColor, Flags )
```

The flags are special functions like ToolExtendLeft, ToolExtendTop etc.

```

Input:
LongColor( ColorGreen ),
FlatColor( Transparent ),
ShortColor( ColorRed );

Variables:
color,
flags( ToolDrawInBackground + ToolExtendBottom + ToolExtendLeft + ToolExtendRight +
ToolExtendTop );

// Select the background color
If MarketPosition = MarketPositionLong Then
farbe = LongColor
Else If MarketPosition = MarketPositionFlat Then
farbe = FlatColor
Else If MarketPosition = MarketPositionShort Then
farbe = ShortColor;

// Remove all previously drawn rectangles
While ( ToolGetFirst() <> -2 )
ToolDelete( ToolGetFirst() );

// Draw a rectangle in the background
DrawRectangle( Date, Time, Close, Date, Time, Close, color, color, flags );

```

CHANGE THE CHART TYPE OF AN INDICATOR/STRATEGY

You can change the output of an indicator or strategy in a chart (which is using one of the drawing functions described above) without changing or recompiling the script. For this, select the indicator or strategy and open the Chart Types button menu in the *Chart* group in the toolbar. Then select the new chart type from the list.

SMALL EXCURSION IN CHROMATICS

In Equilla, there are several ways to enter colors. A number of color definitions can be entered as names. The usual ones are the standard colors like red, darkgreen, blue, black, magenta etc. The list of predefined colors offers many more colors.

Beyond that, the two color models HSV and RGB are available. When you define a color, for example in the color dialog in the property inspector, the RGB and HSV values are given below the color areas. With these values, every possible color can be entered.

THE TWO COLOR MODELS

```
//Output in a subchart
Meta:
    Subchart( true );

//Color values can be saved in variables
Variables:
    myFirstRGBColor, myFirstHSVColor;

//Color as RGB value
myFirstRGBColor = RGB( 100, 30, 250 );

//Color as HSV value
myFirstHSVColor = HSV( 100, 230, 250 );

//Lines in RGB color
DrawLine( 100, "RGB color", StyleSolid, 5, myFirstRGBColor );

//Lines in HSV color
DrawLine( -100, "HSV color", StyleSolid, 5, myFirstHSVColor );
```

USING OPACITY (TRANSPARENCY)

For every color, you can define an opacity value. This works for all colors, the standard colors as well as colors given by RGB or HSV values. Once again, color values can be given by variables which can be combined with opacity.

```
//Open a subchart
Meta:
    Subchart( true );

//Change this input value to see the opacity
//effect in the chart
Inputs:
    Transparency( 200, 1 );

//Color settings can be given as variables
Variables:
    myFirstRGBColor, MyFirstTransparentColor;

//Compose an RGB value
myFirstRGBColor = RGB( 100, 30, 250 );
```

```
//Add Opacity
myFirstTransparentColor = TransparentColor( myFirstRGBColor, Transparency );

//Draw an area with a transparent fill color
DrawArea( 100, -100, "Up", "Down", myFirstTransparentColor, black, black );
```

OUTPUT OF DATA IN THE OUTPUT WINDOW

The output window is used for the output of data or text with the *Print* statement. For example, you can output your own statistical or temporary results.

As a developer, it is often necessary to have detailed information about the program flow. With the *Print* function, a parameter output can be generated (emulating a debug mode) that helps finding eventual errors.

```
//Let the program communicate
Variables:
    myDate, myFutureDate;

myDate = FormatDate( DateTime, "dd.mm.yyyy");
myFutureDate = FormatDate( DateTime, "dd.mm.yyyy")[-1];

Print( "I'm just editing the: " + myDate + ", after that, " +
myFutureDate + " will be next.");

/* In the output window, the following appears: I'm just editing the: 2007/06/12, after
that, 2007/06/14 will be next. */
```

DATA OUTPUT VIA FILE INTERFACE

Tradesignal offers the option to write data directly into a file. You can find the setting for the target folder of the PrintToFile function in the Tradesignal Options, tab *Equilla*. In the statement, only the file name can be specified.

```
Variables:
    fileName("UserStat.txt");

if isLastBar Then
    PrintToFile( fileName, "Indicator was last used at: " +
FormatDate( DateTime, "yyyy/mm/dd"));
/* A file is created with the entry: 'Indicator was last used at 2006/04/30' */
```

As long as you do not change or delete the file, Tradesignal appends each new line to the end of the file. This way, a history of entries results. If you want to ensure by way of a script that an existing file is used, you can delete that file.

```
Variables:
    fileName("UserStat.txt");

if isLastBar Then
    Begin
        //RemoveFile deletes the file
        RemoveFile( fileName );
        PrintToFile( fileName, "Indicator was last used at: " +
FormatDate( DateTime, "yyyy/mm/dd" ));
    End;

/* A file is created with the entry: 'Indicator was last used at 2006/04/30' At each start
of the indicator, the existing file is deleted first. */
```

ALERT OUTPUT

General information on how to set up and use the alert function can be found in Alerts. You also have the option to output alerts via Equilla. Please note that an alert is only output if generated at the last bar of the chart. Therefore it does not matter at which position in the script the alert function is placed, since it is only executed when the last bar is reached.

```
Variables:
    myAlertText("");

If isLastBar Then
    Begin
        myAlertText = "The prices break down!"
        alert( myAlertText );
    End;
```



An Alert Window

EQUILLA ARRAYS

The following article offers an introduction to arrays in Equilla.

Arrays in Equilla are similar to those in other programming languages. They are data fields designed to contain large quantities of data. Each piece of data can be retrieved via the unique position (index) in the array. With the help of additional functions delivered with Tradesignal, a wide range of applications can be covered with arrays.

Arrays are set to replace the Equilla Vectors of earlier Tradesignal versions. Any current script with vectors can easily be modified to use arrays instead.

WHAT IS AN ARRAY?

In the basic Equilla article, variables were introduced as being "containers with an identification number". Via the name of the variable, existing contents can be called up or new contents can be assigned to the variable.

While you can only save one value per variable, arrays are a lot more flexible - an infinite number of values can be put into an array. To expand the container comparison: arrays are containers that include an unlimited number of storage spaces, like boxes in a container. Each box can be retrieved by its index (unique position in the array).

By taking this container analogy one step further to a stack of containers, we get to multidimensional arrays, most commonly described as arrays of arrays.

ARRAY DECLARATION

An array is treated similar to a variable. They are declared in a special declaration block starting with the keyword *Arrays*. The name of the array variable has to be followed by a pair of square brackets.

```
Arrays:  
    myVeryFirstArray[];
```

This array is initially empty and can later be resized to hold as many items as you wish to store in it.

If you know in advance how many items you want to store in your array and don't want to be able to resize it at a later time, you can declare it in this way.

```
Arrays:  
    myFixedSizeArray[10];
```

This creates an array with a maximum valid index of 10, but since indices start at 0, you can store 11 items in this array. Builtin functions ignore index 0, so you should not use this place if you plan to use the builtin functions.

The declaration of a multidimensional array looks like this:

Arrays:

```
my2DArray[3,4];
```

This array can be interpreted as a spreadsheet having rows and columns.

```
[0,0] [0,1] [0,2] [0,3] [0,4]
```

```
[1,0] [1,1] [1,2] [1,3] [1,4]
```

```
[2,0] [2,1] [2,2] [2,3] [2,4]
```

```
[3,0] [3,1] [3,2] [3,3] [3,4]
```

A builtin function would ignore any index containing a 0, but you are free to use these items to store e.g. sums per row and column.

As with variables, you can create a global array. This way it is possible to call up the array from several scripts. Global array are defined just like global variables by adding a namespace for your global variables and :: in front of the array name.

Arrays:

```
myGlobals::myGlobalArray[];
```

ACCESSING ITEMS IN AN ARRAY

The storage places inside an array are addressed using the array name followed by the unique index in square brackets.

Arrays:

```
myArray[10];
// assigns 23 to the storage place with index 1
myArray[1] = 23;
// prints the value contained in storage place with index 1, i.e. 23
Print( myArray[1] );
```

Arrays can include all data types available in Equilla.

OUTPUT CATENATED DATA

If you want to output the complete content of a array in a quick and easy way, you can use several builtin functions. You can just use the print command and all items stored in the array will be printed.

Arrays:

```
myArray[];
// fill the array with some values
FillArray( myArray, 12, 4, 2, 23 );
// output all values using the print command
Print( myArray );
// Print( myArray ) does the same as
// Print( myArray[1], myArray[2], myArray[3], myArray[4] )
```

In the output window appears

```
12      4      2      23
```

Or you can use Join() and have all items printed on a separate line using the string constant NewLine.

```
Arrays:
myArray[];
// fill the array with some values
FillArray( myArray, 12, 4, 2, 23 );
// output all values separated by a newline
Print( Join( NewLine, myArray ) );
```

Output window shows

```
12
4
2
23
```

READING THE ARRAY PROPERTIES

Some useful functions are available for reading the array properties. For example, by using the *ArrayLength()* function, you receive the array size, i.e. the number of items in it.

With other functions you gain information about the highest or lowest value in an array (and their unique index in the array).

- *ArrayLength(ArrayName)* - Returns the size of the array.
- *HighestArray(ArrayName)* - Returns the highest value in the array.
- *IndexOfHighestArray(ArrayName)* - Returns the index of the highest value in the array.
- *LowestArray(ArrayName)* - Returns the lowest value in the array.
- *IndexOfLowestArray(ArrayName)* - Returns the index of the lowest value in the array.

With the following little routine you can test the functions listed above. When you change the content of the *FillArray()* function, different values will be output.

```

Variables:
    OutputText;
Arrays:
    myArray[];
If isLastBar Then
    Begin
        //Fill the array with values
        FillArray( myArray, 2, 4, 6, 8, 10, 2, 3, 5 );
        //Output array contents
        Print( "Array Contents: " + Join( ", ", myArray ) );
        //Read the number of elements
        OutputText = "Number of Elements: " + ArrayLength( myArray );
        //Output in the output window
        Print( OutputText );
        //Retrieve the highest value
        OutputText = "Highest Value: " + HighestArray( myArray );
        //Retrieve unique ID / position of highest value
        OutputText = OutputText + " - at position No.: " +
        IndexOfHighestArray( myArray );
        //Output in the output window
        Print( OutputText );
        //Retrieve the lowest value
        OutputText = "Lowest Value: " + LowestArray( myArray );
        //Retrieve unique ID / position of lowest value
        OutputText = OutputText + " - at position No.: " +
        IndexOfLowestArray( myArray );
        //Output in the output window
        Print( OutputText );
    End;

```

In the output window, the following appears:

```

Array contents: 2, 4, 6, 8, 10, 2, 3, 5
Number of elements: 8
Highest value: 10 - at position No.: 5
Lowest value: 2 - at position No.: 1

```

OVERVIEW OF ALL ARRAY FUNCTIONS

In the following you can find a list of all array functions in Equilla. More information and examples for each function you can find in the Tradesignal help under **Equilla Formula Language**, menu entry **Arrays**.

To open the Tradesignal help, click on the button with the yellow question mark in the toolbox or in the header bar. In the Equilla editor, you can also right-click on an Equilla function and select **Lookup Equilla Function**.

Function	Description
AppendArray	Appends an array or a single variable to another array.
ArrayLength	Returns the length (and maximum index) of an array.
Array_Copy	Copies some elements from one array to another.
Array_GetType	Returns the type of the array elements.
Array_Sort	Sorts a range of elements in an array.
Array_Sum	Returns the sum of all elements or the number of true values in case of booleans.
AverageArray	Returns the (mean) average of all or some elements of the array.
AvgDeviationArray	Returns the average deviation of all or some elements of the array.
BinarySearchArray	Returns the position of a given value using a binary search.
ClearArray	Clears an array and resets it to its initial state.
CompareArray	Compares one array to another.
CopyArray	Copy all or some elements from one array to another.
CreateSortedIndexArray	Creates a sorted index into another array.
FillArray	Fills an array with the given values.
GetArrayDimensions	Returns the number of dimensions of an array.
HarmonicMeanArray	Returns the harmonic mean of all or some elements of the array.
HighestArray	Returns the highest value in the array.
IndexOfHighestArray	Returns the index of the highest value in the array.
IndexOfLowestArray	Returns the lowest value in the array.
InsertAtArray	Inserts a value into an array at a given index.
InsertSortedArray	Inserts a value into an array keeping it sorted.
KurtosisArray	Returns the Kurtosis value of all or some elements of the array.
LowestArray	Returns the lowest value in the array.
MedianArray	Returns the median of all or some elements of the array.
RemoveAtArray	Removes an element at a given index from the array.
SearchArray	Returns the position of a given value.
SetArrayLength	Shrinks or grows a dynamic array.
SetValRangeArray	Sets a range of elements to a given value.
SkewnessArray	Returns the skewness of all or some elements of the array.
Sort2DArray	Sorts a 2-dimensional array by columns.
SortArray	Sorts some or all elements of an array.
StdDeviationArray	Returns the standard deviation of all or some elements of the array.
StdErrorArray	Returns the standard error of all or some elements of the array.
SummationArray	Returns the sum of all or some elements of the array.
SummationRecArray	Returns the sum of reciprocals of all or some elements of the array.

SummationSqrArray	Returns the sum of squares of all or some elements of the array.
VarianceArray	Returns the variance of all or some elements of the array.

EQUILLA VECTORS

The following article offers an introduction to vectors in Equilla. Vectors are deprecated in Tradesignal and are superseded by arrays.

Vectors in Equilla are the same kind of construction as those called arrays in other programming languages. They are data fields designed to contain large quantities of data. Each piece of data can be retrieved via the unique position (index) in the vector. With the help of additional functions delivered with Tradesignal, a wide range of applications can be covered with vectors.

WHAT IS A VECTOR?

In the basic Equilla article, variables were introduced as being "containers with an identification number". Via the name of the variable, existing contents can be called up or new contents can be assigned to the variable.

While you can only save one value per variable, vectors are a lot more flexible - an infinite number of values can be put into a vector. To expand the container comparison: vectors are containers that include an unlimited number of storage spaces, like boxes in a container. Each box can be retrieved by its index (unique position in the vector).

VECTOR DECLARATION

A vector is treated similar to a variable. In the variable declaration, the vector has to be defined and named. The keyword *Vector* has to be added in brackets.

```
Variables:
    myVeryFirstVector( Vector );
```

When you forget the keyword *Vector*, a variable of this name is available but without the vector properties.

As with variables, you can create a global vector. This way it is possible to call up the vector from several scripts. Global vectors are defined as follows:

```
Variables:
    myVeryFirstGlobalVector( GlobalVector );
```

READ AND WRITE DATA

Vectors offer the two basic features *ReadVector()* and *WriteVector()*. Both functions need the name and the unique index to address the right storage place. When writing a vector, you also need to pass the content to be written.

Vectors can include all data types available in Equilla.

WRITE DATA

Data is written into the vector with the *WriteVector()* function. As parameters, the vector name and the value to be written have to be passed. The size of the vector is automatically adapted to the highest used index. Therefore, the possible length of the future vector is irrelevant for the calculation.

```
Variables:
    myVeryFirstVector( Vector );

If isLastBar Then
    Begin
        //The string 'Hello' is written at the position with ID 0
        WriteVector( myVeryFirstVector, 0, "Hello");
        //The string 'Rene' is written at the position with ID 1
        WriteVector( myVeryFirstVector, 1, " Rene");
    End;
```

OUTPUT CATENATED DATA

If you want to output the complete content of a vector in a quick and easy way, you can use the function *FormatVector*. It is part of the delivered functions and concatenates all elements of a vector to one string. This can then be output with the print command. The function is especially useful if you want to keep track of the values while programming.

We expand the source code given above:

```
Variables:
    myVeryFirstVector( Vector );

If isLastBar Then
    Begin
        //The string 'Hello' is written at the position with ID 0
        WriteVector( myVeryFirstVector, 0, "Hello");
        //The string 'Rene' is written at the position with ID 1
        WriteVector( myVeryFirstVector, 1, " Rene");
        Print( FormatVector( myVeryFirstVector ));
    End;
```

In the output window, the line "Hello Rene" appears.

READ DATA

Vector data can be read with the *ReadVector()* function. As parameters, the vector name and the index have to be passed.

Make sure that the program does not try to read from non-existing positions. When this happens, Tradesignal aborts the execution with an error message in the chart legend.

```
Variables:
    myVeryFirstVector( Vector ), OutputText;

If isLastBar Then
    Begin
        //The string 'Hello' is written at the position with ID 0
        WriteVector( myVeryFirstVector, 0, "Hello");
        //The string 'Rene' is written at the position with ID 1
        WriteVector( myVeryFirstVector, 1, " Rene");
        OutputText = ReadVector( myVeryFirstVector, 0 );
        OutputText = OutputText + ReadVector( myVeryFirstVector, 1 );
        Print( OutputText );
    End;
```

In the output window, the line "Hello Rene" appears.

READING THE VECTOR PROPERTIES

Some useful functions are available for reading the vector properties. For example, by using the *VectorLength()* function, you receive the vector size, i.e. the number of included elements.

With other functions you gain information about the highest or lowest value in a vector (and their unique IDs in the vector). Please note that a vector indexing always starts at zero. The first index is zero, the second is one and so on.

- *VectorLength(VectorName)* - Offers the number of elements.
- *HighestVector(VectorName)* - Offers the highest value in the vector as result.
- *IndexOfHighestVector(VectorName)* - Offers the unique ID of the highest value in the vector.
- *LowestVector(VectorName)* - Offers the lowest value in the vector as result.
- *IndexOfLowestVector(VectorName)* - Offers the unique ID of the lowest value in the vector.

With the following little routine you can test the functions listed above. When you change the content of the *FillVector()* function, different values will be output.

```
Variables:
    myVeryFirstVector ( Vector ), OutputText;
```

```

If isLastBar Then
    Begin
        //Fill the vector with values
        FillVector( myVeryFirstVector , 2,4,6,8,10,2,3,5 );
        //Output vector contents
        Print( "Vector Contents: " + FormatVector( myVeryFirstVector ));
        //Read the number of elements
        OutputText = "Number of Elements: " + VectorLength( myVeryFirstVector );
        //Output in the output window
        Print( OutputText);
        //Retrieve the highest value
        OutputText = "Highest Value: " + HighestVector( myVeryFirstVector );
        //Retrieve unique ID / position of highest value
        OutputText = OutputText + " - at position No.: " +
        IndexOfHighestVector( myVeryFirstVector );
        //Output in the output window
        Print( OutputText);
        //Retrieve the lowest value
        OutputText = "Lowest Value: " + LowestVector( myVeryFirstVector );
        //Retrieve unique ID / position of lowest value
        OutputText = OutputText + " - at position No.: " +
        IndexOfLowestVector( myVeryFirstVector );
        //Output in the output window
        Print( OutputText);
    End;

```

In the output window, the following appears:

```

Vector contents: '2', '4', '6', '8', '10', '2', '3', '5'
Number of elements: 8
Highest value: 10 - at position No.: 4
Lowest value: 2 - at position No.: 0

```

FILL VECTORS WITH CONTENT AND DELETE CONTENT

To write a defined number of values into a vector, use the function *FillVector(VectorName, Value1, Value2, ...)*. This function was already used in the example above.

To delete the content of a vector, use the function *ClearVector(VectorName)*. This function deletes all elements in a vector and resets its size to zero. Use this function when you reuse vectors in your program, except in cases when you want to use old vector data.

Let us expand the source code of the example above by adding the delete function:

Variables:

```
myVeryFirstVector ( Vector ), OutputText;
```

```
If isLastBar Then
```

```
Begin
```

```
    //Fill the vector with values
```

```
    FillVector( myVeryFirstVector , 2,4,6,8,10,2,3,5 );
```

```
    //Output vector contents
```

```
    Print( "Vector Contents: " + FormatVector( myVeryFirstVector ));
```

```
    //Read the number of elements
```

```
    OutputText = "Number of Elements: " + VectorLength( myVeryFirstVector );
```

```
    //Output in the output window
```

```
    Print( OutputText);
```

```
    //Retrieve the highest value
```

```
    OutputText = "Highest Value: " + HighestVector( myVeryFirstVector );
```

```
    //Retrieve unique ID / position of highest value
```

```
    OutputText = OutputText + " - at position No.: " +
```

```
    IndexOfHighestVector( myVeryFirstVector );
```

```
    //Output in the output window
```

```
    Print( OutputText);
```

```
    //Retrieve the lowest value
```

```
    OutputText = "Lowest Value: " + LowestVector( myVeryFirstVector );
```

```
    //Retrieve unique ID / position of lowest value
```

```
    OutputText = OutputText + " - at position No.: " +
```

```
    IndexOfLowestVector( myVeryFirstVector );
```

```
    //Output in the output window
```

```
    Print( OutputText);
```

```
    Print( "Vector deleted!" );
```

```
    ClearVector( myVeryFirstVector );
```

```
    Print( "Vector contents: " + FormatVector( myVeryFirstVector ));
```

```
    //Repeat all analysis:
```

```
    OutputText = "Number of Elements: " + VectorLength( myVeryFirstVector );
```

```
    //Output in the output window
```

```
    Print( OutputText);
```

```
    //Retrieve the highest value
```

```
    OutputText = "Highest Value: " + HighestVector( myVeryFirstVector );
```

```
    //Retrieve unique ID / position of highest value
```

```
    OutputText = OutputText + " - at position No.: " +
```

```
    IndexOfHighestVector( myVeryFirstVector );
```

```
    //Output in the output window
```

```
Print( OutputText);
//Retrieve the lowest value
OutputText = "Lowest Value: " + LowestVector( myVeryFirstVector );
//Retrieve unique ID / position of lowest value
OutputText = OutputText + " - at position No.: " +
IndexOfLowestVector( myVeryFirstVector );
//Output in the output window
End;
```

In the output window, the following appears:

```
Vector contents: '2', '4', '6', '8', '10', '2', '3', '5'
Number of elements: 8
Highest value: 10 - at position No.: 4
Highest value: 2 - at position No.: 0
Vector deleted!
Vector contents:
Number of elements: 0
Highest value: 0 - at position No.: -1
Highest value: 0 - at position No.: -1
```

As you can see, the vector is empty after the deletion. Due to this, some functions output an error with the result "-1".

FINDING AND SORTING DATA

Some useful functions are available to navigate in a vector.

To find data in a vector, use the function *SearchVector(VectorName, SearchedValue)*.

To sort data, use the function *SortVector(VectorName, SortAscending)*.

To demonstrate the sorting, we modify the indicator of the deletion example. We simply exchange the *ClearVector(VectorName)* function with the sorting function *SortVector(VectorName, SortAscending)*. You can change the sorting direction with true/false. Enter true for ascending and false for descending order.

```
Variables:
myVeryFirstVector ( Vector ), OutputText;

If isLastBar Then
Begin
```

```
//Fill the vector with values
FillVector( myVeryFirstVector , 2,4,6,8,10,2,3,5 );
//Output vector contents
Print( "Vector Contents: " + FormatVector( myVeryFirstVector ));
//Read the number of elements
OutputText = "Number of Elements: " + VectorLength( myVeryFirstVector );
//Output in the output window
Print( OutputText);
//Retrieve the highest value
OutputText = "Highest Value: " + HighestVector( myVeryFirstVector );
//Retrieve unique ID / position of highest value
OutputText = OutputText + " - at position No.: " +
IndexOfHighestVector( myVeryFirstVector );
//Output in the output window
Print( OutputText);
//Retrieve the lowest value
OutputText = "Lowest Value: " + LowestVector( myVeryFirstVector );
//Retrieve unique ID / position of lowest value
OutputText = OutputText + " - at position No.: " +
IndexOfLowestVector( myVeryFirstVector );
//Output in the output window
Print( OutputText);

SortVector( myVeryFirstVector , true );

//Repeat the analysis:
Print( "Vector contents: " + FormatVector( myVeryFirstVector ));
//Fill the vector with values
FillVector( myVeryFirstVector , 2,4,6,8,10,2,3,5 );
//Output vector contents
Print( "Vector Contents: " + FormatVector( myVeryFirstVector ));
//Read the number of elements
OutputText = "Number of Elements: " + VectorLength( myVeryFirstVector );
//Output in the output window
Print( OutputText);
//Retrieve the highest value
OutputText = "Highest Value: " + HighestVector( myVeryFirstVector );
//Retrieve unique ID / position of highest value
OutputText = OutputText + " - at position No.: " +
IndexOfHighestVector( myVeryFirstVector );
//Output in the output window
Print( OutputText);
//Retrieve the lowest value
OutputText = "Lowest Value: " + LowestVector( myVeryFirstVector );
//Retrieve unique ID / position of lowest value
```

```

        OutputText = OutputText + " - at position No.: " +
IndexOfLowestVector( myVeryFirstVector );
    //Output in the output window
    Print( OutputText);
End;

```

In the output window, the following appears:

```

Vector contents: '2', '4', '6', '8', '10', '2', '3', '5'
Number of elements: 8
Highest value: 10 - at position No.: 4
Lowest value: 2 - at position No.: 0
Vector contents: '2', '2', '3', '4', '5', '6', '8', '10'
Number of elements: 8
Highest value: 10 - at position No.: 7
Lowest value: 2 - at position No.: 0

```

Here is an example for the search function. The wanted value is "10".

```

Variables:
    myVeryFirstVector ( Vector )

If isLastBar Then
    Begin
        //Fill the vector with values
        FillVector( myVeryFirstVector , 2,4,6,8,10,2,3,5 );
        //Search the vector for a value
        Print( "The wanted value 10 is at position No.: " +
SearchVector( myVeryFirstVector, 10 ));
    End;

```

In the output window appears:

```

Vector content: '2', '4', '6', '8', '10', '2', '3', '5'
The wanted value 10 is at position No.: 4

```

OVERVIEW OF ALL VECTOR FUNCTIONS

In the following you can find a list of all vector functions in Equilla. More information and examples for each vector you can find in the Tradesignal help under **Equilla Formula Language**, menu entry **Vectors**.

To open the Tradesignal help, click on the button with the yellow question mark in the toolbox or in the header bar. In the Equilla editor, you can also right-click on an Equilla function and select **Lookup Equilla Function**.

Vector	Funktion
AppendVector()	Appends the contents of one vector onto the end of another vector.
AverageVector()	Returns the average value (mean) of the elements in a vector.
AvgDeviationVector()	Returns the average deviation of the elements in a vector.
BinarySearchVector()	Returns the position of an element with a specific value in a vector using a binary search.
ClearVector()	Clears the contents of a vector and sets its size to zero (0).
CompareVector()	Compares the elements in one vector to those in another vector.
CopyVector()	Copies some or all of the elements from one vector to another.
CreateSortedIndexVector() ()	Creates a sorted index to a vector. The vector containing the actual data will remain unchanged.
FillVector()	Fills a vector with the specified elements.
GlobalVector()	Returns a newly created vector that can be shared between scripts.
HarmonicMeanVector()	Returns the harmonic mean of the values in a vector.
HighestVector()	Returns the highest element in a vector.
IndexOfHighestVector()	Returns the zero based index of the first occurrence of the highest element in a vector.
IndexOfLowestVector()	Returns the zero based index of the first occurrence of the lowest element in a vector.
KurtosisVector()	Returns the Kurtosis value of the elements in a vector.
LowestVector()	Returns the lowest element in a vector.
MedianVector()	Returns the median of the elements in a vector.
ReadVector()	Returns the value stored at a specific index within a vector.
ReadVectorIndirect()	Returns the value stored at a certain index within the data vector.
SearchVector()	Returns the position of an element with a specific value in a vector.
SetMaxIndexVector()	Sets the largest index value for this vector.
SetValRangeVector()	Sets a range of elements in a vector to a specified value.
SkewnessVector()	Returns the skewness (the degree of asymmetry) of the elements in a vector.
SortVector()	Sorts the contents of the vector.
StdDeviationVector()	Returns the standard deviation of the elements in a vector.
StdErrorVector()	Returns the standard error of the elements in a vector.
SummationRecVector()	Returns the sum of reciprocals of each element in a vector.
SummationSqrVector()	Returns the sum of squares of each element in a vector.
SummationVector()	Returns the sum of all, or a range of elements in the vector.
VarianceVector()	Returns the variance of the elements in a vector.
Vector()	Returns a newly created vector.
VectorLength()	Returns the number of elements stored in a vector..

WriteVector()	Writes a value to a specific index within a vector.
WriteVectorIndirect()	Writes a value to a specific index within a vector. The index into the data vector will be looked up via the index vector.

EXTENDING EQUILLA

Tradesignal offers two ways of extending the equilla programming language.

C-API EXTENSIONS

C-API extensions are supported for backwards compatibility. They work by exporting functions from dlls that are placed in a special folder. These functions can be referenced from equilla using the *Import* statement. For a detailed documentation read the article C-API Extensions.

COM-API EXTENSIONS

These Extensions have been introduced with Tradesignal 6.3 and are the preferred way of extending equilla. You can implement these extensions using programming languages like C#, C++ or any other language that can output COM compatible dlls. COM Extensions allow an *Object* style syntax to access the implemented methods. For a detailed documentation read the article COM-API Extensions.

INTRODUCTION TO FUNCTIONS, INDICATORS AND STRATEGIES

The hub of all analysis methods in Tradesignal is the available functions, indicators and strategies.

All three tools are written in Equilla Code. This is a Tradesignal-specific programming language similar to Algol or Visual Basic. No programming knowledge is necessary as long as you use the delivered functions, indicators or strategies. However, if you are interested in programming, see the Equilla Basics chapters for details.

TOOL CHARACTERISTICS

FUNCTIONS

- Functions are subordinate portions of the code in which regularly used calculations and subroutines are sourced out. For example, instead of writing the code for the standard deviation again and again, the calculation is put into a function. By calling up the function by its name - and passing the relevant parameter(s) for calculation - the user receives the return value as a result.
- A function cannot be applied to a chart directly.

You can find more information in the chapter Equilla Functions.

INDICATORS

- Indicators are mathematical calculations that can be applied to symbol data for technical analysis.
- Indicators can be applied to charts or other indicators, watchlists, portfolios etc.

You can find more information in the chapter Using Indicators.

STRATEGIES

- Strategies combine the results of indicators with certain rules and conditions for signal generation, to help setting up a sophisticated trading system.
- Strategies can be applied to charts, indicators, watchlists, portfolios etc.

You can find more information in the chapter Using Strategies.

EXAMPLE VOLATILITY

- The function "Volatility" calculates the standard deviation from the moving average of the close price.
- The indicator "Volatility Ratio" uses the Volatility function over two different periods and then calculates the ratio of the two results. A Volatility Ratio above 0.5 signals a breakout.
- The strategy "Volatility Ratio Breakout - Exit" combines the indicator with an Ehlers filter, so that long or short positions are

closed depending on the development of the volatility.

EXAMPLE BOLLINGER BANDS

- The function "BollingerBands" calculates a simple moving average and an upper and lower standard deviation as bands.
- The indicator "Bollinger Bands" calculates and draws the resulting bands. In the property inspector of the toolbox, the parameters price (close, high...), period, and standard deviation can be edited.
- The strategy "Bollinger Lower Band - Entry" combines the results of the indicator with entry signal generation.

EQUILLA CODE OF THE BOLLINGERBANDS FUNCTION

Meta:

Synopsis("[Bollinger Bands] Calculates the Bollinger band values for the upper and lower deviation bands and the simple moving average and returns them as output parameters. Bollinger Bands are an indicator that allows users to compare volatility and relative price levels over a period time. The three calculated values are designed to encompass the majority of a security's price action. Sharp price increases or decreases (volatility) will lead to a widening of the band. Consolidation will result to a thinning of the bands.");

Inputs:

```
Price( NumericSeries ),
Period( NumericSimple ),
StdDevs( NumericSimple ),
RefMidBand( NumericRef ),
RefUpperBand( NumericRef ),
RefLowerBand( NumericRef );
```

Variables:

```
distance, sumSqr, i;
```

```
RefMidBand = AverageFC( Price, Period );
```

```
sumSqr = 0;
```

```
For i = 0 To Period - 1 Do
    sumSqr = sumSqr + Sqr( Price[i] - RefMidBand );
```

```
If sumSqr > 0 And Period > 0 Then
    distance = Sqrt( sumSqr / Period ) * StdDevs
```

```
Else
    distance = 0;
```

```
RefUpperBand = RefMidBand + distance;
```

```
RefLowerBand = RefMidBand - distance;

BollingerBands = 1;

// *** Copyright Tradesignal GmbH ***
// *** www.tradesignal.com ***
```

EQUILLA CODE OF THE BOLLINGERBANDS INDICATOR

Meta:

```
Synopsis( "Bollinger Bands are an indicator that allows users to compare volatility and
relative price levels over a period time. The indicator consists of a moving average and two
bands drawn two standard deviations from the average. These three bands are designed to
encompass the majority of a security's price action. Sharp price increases or decreases
(volatility) will lead to a widening of the band. Consolidation will result to a thinning of
the bands." ),
ShortCode( "BBD" ),
SubChart( False );
```

Inputs:

```
Price( Close ),
Period( 20, 1 ),
StdDevs( 2.0, 0.0 );
```

Variables:

```
avg, lowerBand, upperBand;
```

```
BollingerBands( Price, Period, StdDevs, avg, upperBand, lowerBand );
```

```
DrawLine( avg, "Mid Line", StyleDot );
DrawArea( upperBand, lowerBand, "Upper Band", "Lower Band" );
```

```
// *** Copyright Tradesignal GmbH ***
// *** www.tradesignal.com ***
```

EQUILLA CODE OF THE "BOLLINGER LOWER BAND - ENTRY" STRATEGY

Meta:

```
Synopsis( "Generates a long entry signal when the Close crosses over the Bollinger Lower
Band value, and/or generates a short entry signal when the Close crosses under the Bollinger
Lower Band value. Bollinger Bands are an indicator that allows users to compare volatility
and relative price levels over a period time. The indicator consists of a moving average and
two bands drawn two standard deviations from the average. These three bands are designed to
```

encompass the majority of a security's price action. Sharp price increases or decreases (volatility) will lead to a widening of the band. Consolidation will result to a thinning of the bands.");

Inputs:

```
Price( Close ),
Period( 20, 1 ),
StdDevs( 2.0, 0.0 ),
EntryMethod ( LongEntry, ShortEntry, Both ) = Both,
Visuals( False );
```

Variables:

```
longSig, shortSig, avg, upperBand, lowerBand;
```

```
BollingerBands( Price, Period, StdDevs, avg, upperBand, lowerBand );
```

```
longSig = ( Close Crosses Over lowerBand ) And ( EntryMethod <> ShortEntry );
```

```
shortSig = ( Close Crosses Under lowerBand ) And ( EntryMethod <> LongEntry );
```

```
If longSig Then
```

```
    Buy( "BollingerLWR" ) Next Bar at Market
```

```
Else If shortSig Then
```

```
    Short( "BollingerLWR" ) Next Bar at Market;
```

```
If Visuals Then
```

```
    DrawLine( lowerBand, "BollingerLowerBand", StyleSolid, 2 );
```

```
// *** Copyright Tradesignal GmbH ***
```

```
// *** www.tradesignal.com ***
```

USING INDICATORS

Technical indicators are mathematical calculations based on securities and additional data. You can use indicators for

- Signal generation in mechanical trading systems
- Detecting trends in symbol charts
- Supporting trading decisions
- Supporting chart analysis

In Tradesignal, indicators are available which were developed based on material from various sources (publications, media or in-house development). For all indicators, additional information on the source, the calculation basis, the interpretation of results and the appliance is offered.

The indicators in Tradesignal are written in Equilla. You can find more information in the Equilla Basics chapters.

APPLYING INDICATORS

APPLYING INDICATORS TO CHARTS OR INDICATORS

You can apply an indicator to a chart (or an indicator) in various ways.

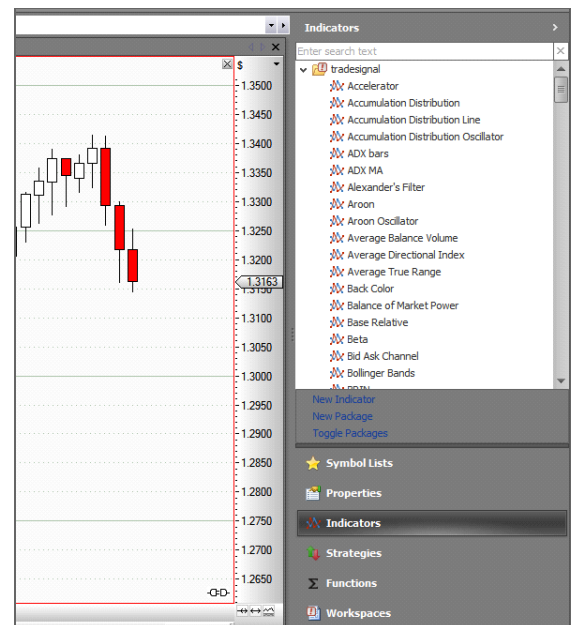
Applying from the indicator list in the toolbox:

1. Open the indicator list in the toolbox by clicking on the **Indicators** button.
2. Apply the indicator:
 1. By drag and drop: Click on the indicator, keep the mouse button pressed and drag the indicator into the chart.
 2. By double-click: Double-click on the indicator.
 3. From the context menu: Right-click on the indicator and choose **Apply** from the context menu.

Applying via the command line in the toolbar:

1. Enter a shortcode in the command line, for example "BBD".
2. Select the **Add Indicator** option in the drop-down menu.

DATA INPUT SELECTION



Detail: Indicator view in the toolbox

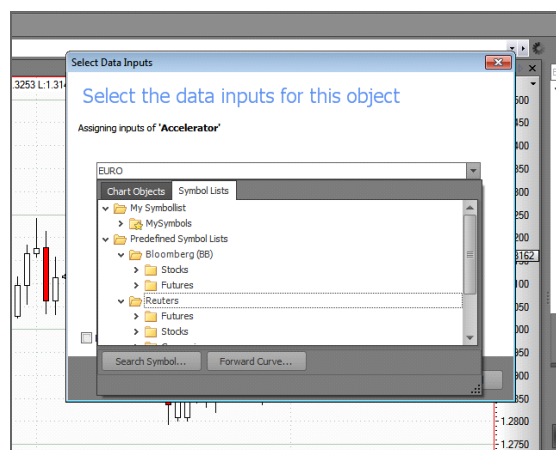


Detail: Entering a shortcode in the commandline

If you apply an indicator which uses more than one data input the data input selection dialog (see image) is displayed. It allows you to select either an already existing chart object or to open a new symbol via the symbol lists or symbol search. There also is an option to always display this dialog so you can apply indicators to symbols which are not themselves displayed.

APPLICATION RULES

- The indicator is applied to the active chart or subchart, except when using drag and drop.
- By default, indicators are applied to the top entry in the chart or subchart legend.
- When you drag the indicator onto a certain entry in the legend of a chart or subchart (for example another indicator), the indicator is applied to this entry.
- Whether a subchart is created for an indicator depends on the meta tag "subchart" in the Equilla Code of the indicator (see the chapter Equilla Program Structure and Syntax).



Data input selection dialog

You can see in the legend how to which entry the indicator is applied. It is constructed as **Indicator[Var1, Var2...] value {applied to}**.

Example:

- Chart: DAX P-IN. [.DAX LAST 1 hour]...
- Subchart#1: Accelerator [Close, 10, 13] 32.02 {.DAX}
- Subchart#2: Average True Range [14] 122.44 {Accelerator}

EXAMPLE: APPLYING BOLLINGER BANDS TO A CHART

1. Open the .NDX in a chart and set the time span to "1 Year" (via the "Fixed Data Range" entry).
2. In the indicator list in the toolbox, search for "Bollinger Bands".
3. Drag the indicator into the graph. The Bollinger Bands are calculated for the .NDX and enfold the line.



Detail: Applying Bollinger Bands to the .NDX

EXAMPLE: APPLYING BOLLINGER BANDS TO AN INDICATOR

A popular combination is the "Relative Strength Index" (RSI) with added Bollinger Bands. Proceed as follows:

1. To your chart, apply the "Relative Strength Index" (RSI) indicator as described above. The result is displayed in a subchart.
2. Select the RSI subchart by clicking into the subchart.
3. Now apply the "Bollinger Bands" (BBD) indicators to the RSI subchart. The Bollinger Bands are calculated for the RSI and enfold its line.

APPLYING INDICATORS TO SCANNER, WATCHLIST OR PORTFOLIO

You can also apply indicators to symbol tables such as the Scanner, Watchlist or Portfolio.

- When creating a new Scanner, Watchlist or Portfolio, you can directly add indicators (and strategies).
- For existing tables, you can add indicators to the selected table by using one of the methods described above. A new column is added for the results.

EXAMPLE: SCANNER WITH INDICATOR

The Scanner is intended for screening a high number of securities for certain criteria. For example, create a Relative Strength list for the hundred Nasdaq values by adding the "Relative Strength Levy" (RLS) to the Scanner and starting the scan. The new column is added and displays the RLS for each instrument.

EXAMPLE: WATCHLIST WITH INDICATOR

The Watchlist is intended for realtime data. For example, drag the Momentum indicator into the table, enter a short trading period and activate the alert function of the indicator. Now Tradesignal issues an alert any time a value in the Watchlist triggers a Momentum signal.

APPLYING INDICATORS VIA THE COMMAND LINE

SHORTCODES

All indicators delivered with Tradesignal feature a shortcode. These are part of the original programming in the Equilla code and should not be changed. Via the command line, an indicator can be applied to Charts, Scanners, Watchlists, and Portfolios. This is faster than applying indicators from the toolbox.

The shortcodes can be looked up in the tooltips of indicators in the toolbox. To display a tooltip, move the mouse over the indicator entry.

EQUILLA CODE

Basically, indicators are constructed so that the calculations are done in functions. An Equilla function can be entered in the Equilla Editor as well as the command line. With a bit of practice, it is possible to enter creative combinations without spending much time on programming. You can find the list of available Equilla functions in the toolbox by clicking on **Functions**. Most names are constructed to make the connections between the indicators and their functions obvious.

EXAMPLE FOR DIRECTLY ENTERING AN INDICATOR

To apply an RSI (smoothed by an Exponential Moving Average), proceed as follows:

1. Open or activate a chart.
2. Enter the following code in the command line:

```
Drawline(XAverage(RSI(Close,14),5))
```

and select the option **Add formula** in the drop-down menu.

You can find more information in the chapter Command Line.

DELETING INDICATORS

DELETING AN INDICATOR FROM THE CHART

There are two ways to delete an indicator from a chart:

- The indicator is displayed in the legend at the top left of the chart or subchart. Right-click on the entry to open the context menu and select **Delete**.
- Alternatively, select the entry in the legend and press the **DEL** key.

To delete a complete subchart of an indicator, click on the X button on the top right. Empty subcharts are closed automatically.

DELETING AN INDICATOR FROM A SCANNER, WATCHLIST OR PORTFOLIO

- To delete an indicator from a table, right-click on the table header and select **Add/Remove Columns**.
- Alternatively, click on the **Columns** button and select **Add/Remove Columns** from the button menu. You can find more information in the chapters Scanner, Watchlist and Portfolio.

REMOVING AN INDICATOR FROM TRADESIGNAL

To delete an indicator from the list in Tradesignal, right-click the indicator in the toolbox and select **Delete** from the context menu. A warning asks you to confirm the deletion.

CREATING A NEW INDICATOR

To create a new indicator, click on the link **New Indicator** in the *Related Tasks* area of the toolbox. A Wizard opens, offering two options:

- **Use the Equilla Formula Language to write the indicator** - Select this option if you want to write the indicator yourself. You can find more information in the chapter Writing Indicators.
- **Download an existing indicator from Tradesignal Online** - Select this option if you want to download a ready-made indicator from the website.

DOWNLOADING AN INDICATOR FROM TRADESIGNAL ONLINE

When you select the second option, the internal web browser opens and the entry "Indikatoren" in the Tradesignal lexicon appears.

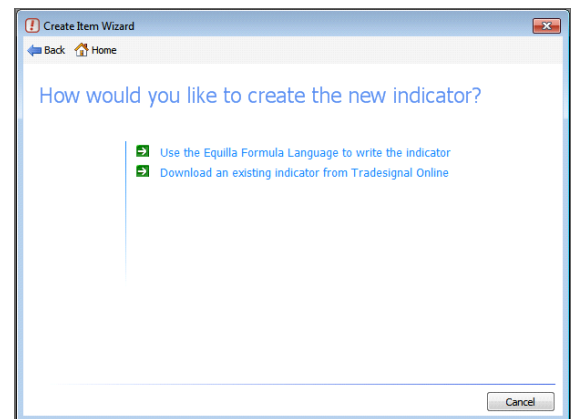
Click on an indicator in the list. The entry for this indicator opens. At the bottom of the page, you will find one or more Equilla scripts (if available) and saving options:

- **Quelltext anzeigen** (show code) – Displays the Equilla code in the internal web browser. You can select and copy this code and, for example, paste it into the Equilla editor in Tradesignal Standard Edition.
- **Skript in Tradesignal Web Edition importieren** (import into Tradesignal Web Edition) – Saves the script in Tradesignal Web Edition. You can then find it in the list of all indicators in the web edition. To save the script in the web edition, you have to be logged into the Tradesignal Online website. In the case of the error message "Not enough space to import the script", you are either not logged in or there is not enough web space left to save the script.
- **Skript in Tradesignal importieren** (import into Tradesignal Standard Edition) – Saves the script in Tradesignal Standard Edition. You can create the indicator in an existing package or create a new package.

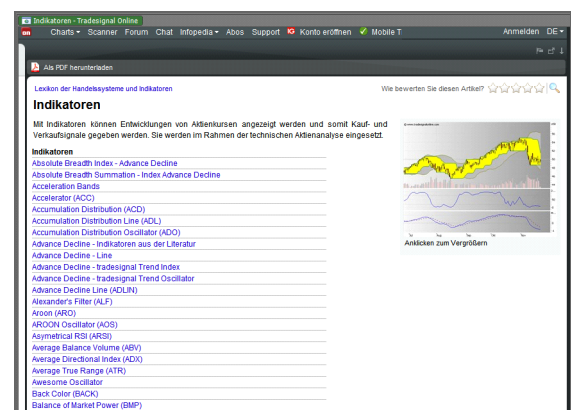
In the case of a new package, the following procedure is recommended:

1. Click on the button **New package**.
2. Select the main package *Tradesignal Files*.
3. Click on **New Folder** and enter the name "Personal Indicators".
4. Confirm all settings by clicking on the **OK** buttons.

EDITING THE EQUILLA CODE OF AN INDICATOR



Indicator Wizard - start screen



Indicator Wizard - download list

- To edit the source code of an indicator, right-click the indicator in the chart legend and select **Edit Equilla Code** from the context menu.
- Alternatively, right-click the indicator in the toolbox and select **Edit** from the context menu.

The source code is opened in the Equilla Editor.

You can find more information in the Equilla chapters.

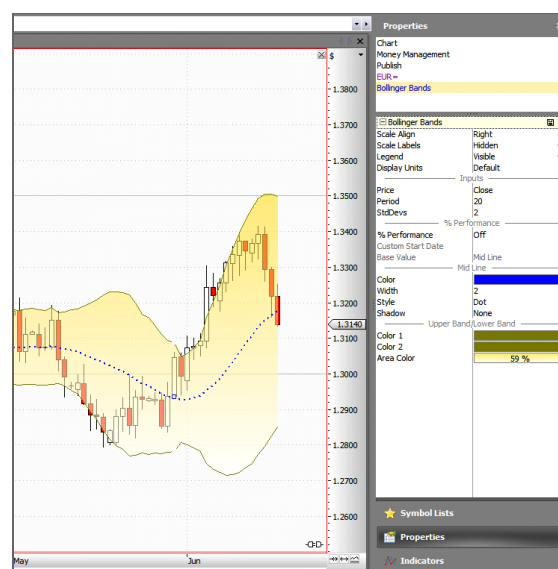
INDICATOR PROPERTIES

The indicator properties are available either from the context menu in the chart or the property inspector in the toolbox. Select the indicator from the list of elements at the top to open the properties.

Each indicator offers various parameters like display units, period, weighting factors, multipliers etc. Often, values for additional scale units like oversold or overbought levels are available. Enter all necessary parameters.

For each element in an indicator (i.e. the indicator or its signal lines, graphic symbols or extrema), display properties can be entered such as line widths, colors or shadows.

For some Tradesignal indicators it is possible to set an alert, which will be created as soon as a defined condition is met, e.g. for SMA (when the Moving Average Simple crosses the main data curve). Alerts will be displayed as a flashing icon in the system tray and as a popup window by default. Furthermore, you can find information about all alerts generated by an indicator in the alert window on the **chart alerts** tab.



Detail: Bollinger Bands Properties

CHANGING THE EVALUATION ORDER OF INDICATORS AND STRATEGIES

In some circumstances, indicators and strategies must be evaluated in a specific order. Normally this is taken care of by Tradesignal, however there are cases where it is not possible to automatically determine which item should go first. For example if multiple indicators use the same Global Variables. By default, indicators and strategies are evaluated in the order they are added to the chart, watchlist, scanner or portfolio. To change this order

1. Click on *Home > Chart > Manage Objects* on the toolbar to show the **Manage Strategies and Indicators dialog**.
2. Click on the item in the list that needs to be evaluated differently.
3. Click on the **Move Up** or **Move Down** buttons to reposition the item. Items closer to the top are evaluated earlier.
4. Click **OK** to commit the changes.

WRITING INDICATORS

In Tradesignal, many indicators are available, either delivered with the software or offered for download from the website.

However, wishes may remain unfulfilled or you may have new ideas that need to be tested. In these cases, you can always program new indicators in Tradesignal by using the Equilla language.

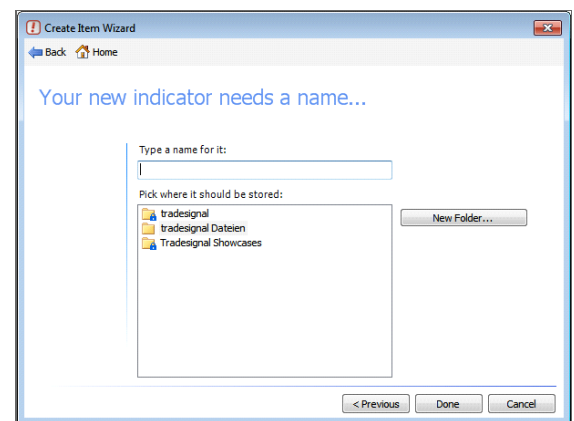
In the following you will find an example showing how to write an indicator using Equilla. It is only intended to give you a basic introduction. For details on Equilla, please refer to the Equilla Basics chapters.

PROGRAMMING TRADING BASED ON A RELATIVE STRENGTH INDEX

CREATING THE INDICATOR

1. In the toolbox, click on the **Indicators** button.
2. In the *Related Tasks* area, select **New Indicator**.
3. In the next dialog, select the option **Use the Equilla Formula Language to write the indicator**.
4. Select a folder, e.g. "Tradesignal Files".
5. Type the name "RSI-Bands".
6. Click on **Done**.

Now you can enter the source code of the indicator in the Equilla Editor.



Writing a new indicator

ENTERING THE META INFORMATION

In the meta area, global information about the indicator and its display are entered. These are declared using meta tags.

```
Meta:
    Subchart ( false ) ;
```

DECLARING THE INPUT PARAMETERS

In the input area, parameters are declared that can later be edited and changed by the user during indicator runtime. In this case, these are mainly a smoothing factor and the calculation period.

```
Inputs:
    Period( 20 , 1 ),
    Smoothing( 10 , 1 ),
    Factor( 1.0 );
```

DECLARING THE VARIABLES

Variables are data containers in which the values of calculation results are temporarily saved and therefore available for future use.

Variables:

```
rsiValue, upperBand, lowerBand, medLine;
```

PERFORMING THE CALCULATIONS

In this area, the actual calculations are done, whose results are displayed later. In our example, the continuous regression line is taken as mid-line. With it, the Relative Strength Index is calculated, which is a measure for the strength and dynamics of the trend. The actual bands result from simply adding and subtracting the RSI from the mid-line.

```
medLine = LinRegValue( Close, Smoothing, 0 );
rsiValue = RSI( Close, Period );
upperBand = medLine + ( rsiValue * factor );
lowerBand = medLine - ( rsiValue * factor );
```

INSERTING THE GRAPH

Finally, the calculation results have to be displayed. This is usually done at the end of the indicator code.

```
DrawLine( medLine, "Average", StyleSolid, 1, black );
DrawLine( upperBand, "Upper", StyleSolid, 2, red );
DrawLine( lowerBand, "Lower", StyleSolid, 2, darkgreen );
```

THE COMPLETE CODE

```
Meta:
    Subchart( false );

Inputs:
    Period( 20 , 1 ),
    Smoothing( 10 , 1 ),
    Factor( 1.0 );

Variables:
    rsiValue, upperBand, lowerBand, medLine;

medLine = LinRegValue( Close, Smoothing, 0 );
```

```
rsiValue = RSI( Close, Period );
upperBand = medLine + ( rsiValue * factor );
lowerBand = medLine - ( rsiValue * factor );

DrawLine( medLine, "Average", StyleSolid, 1, black );
DrawLine( upperBand, "Upper", StyleSolid, 2, red );
DrawLine( lowerBand, "Lower", StyleSolid, 2, darkgreen );
```

APPLYING THE INDICATOR

Compile the code either by pressing **F7** or by clicking on the button **Compile Script** in the *Equilla Editor* group. The code is checked for errors, saved and then available in the list of indicators in the toolbox by the name "RSI-Bands".

You can now apply it to a chart as described in Using Indicators, for example with drag and drop.

USING STRATEGIES

In the traditional sense, trading systems are compilations of rules from which trading decisions of any kind result. You can find more information in the chapter Using Trading Strategies.

One of the important aspects of trading systems is the strategies. Strategies combine indicators with entry and exit rules.

In Tradesignal, strategies are available which were developed based on material from various sources (publications, media or in-house development). For all strategies, additional information on the source, the calculation basis, the interpretation of results and the appliance is offered.

The strategies in Tradesignal are written in Equilla. You can find more information in the Equilla Basics chapters.

APPLYING STRATEGIES

APPLYING STRATEGIES TO CHARTS OR INDICATORS

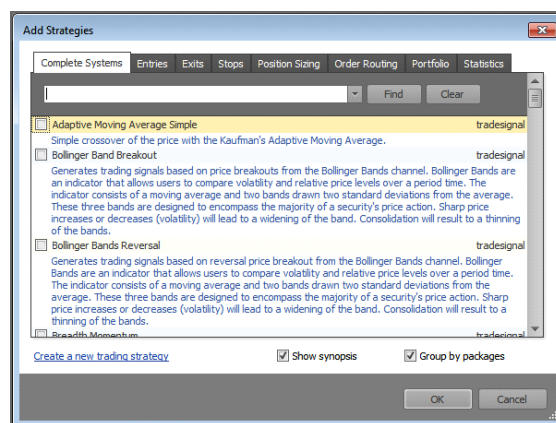
- You can apply a strategy to a chart by clicking on the **Add Strategy** button under *Home > Strategies/Indicators* on the toolbar. This opens a new dialog in which the strategies are sorted in categories and synopses are displayed (if the corresponding option is selected).

- You can also apply a strategy to a chart (or an indicator) from the strategies list in the toolbox:

- Open the strategy list in the toolbox by clicking on the **Strategies** button.
- Apply the strategy:
 - By drag and drop: Click on the strategy, keep the mouse button pressed and drag the strategy into the chart.
 - By double-click: Double-click on the strategy.
 - From the context menu: Right-click on the strategy and choose **Apply** from the context menu.



Window view "Add Strategy"



Detail: Strategy view in the toolbox

Application Rules

- The strategy is applied to the active chart or subchart, except when using drag and drop.
- By default, strategies are applied to the top entry in the chart or subchart legend.
- When you drag the strategy onto a certain entry in the legend of a chart or subchart (for example an indicator), the strategy is applied to this entry.
- Whether a subchart is created for a strategy depends on the meta tag "subchart" in the Equilla Code of the strategy (see the chapter Equilla Program Structure and Syntax).

You can see in the legend how to which entry the strategy is applied. It is constructed as

Strategy[Var1, Var2...] value {applied to}.

Example:

- Chart: DAX P-IN. [.DAX LAST 1 hour]...
- Accelerator - Entry [10, 13, 5] {.DAX}

Since a strategy combines indicators with various conditions, more parameters are possible than for indicators, e.g. for risk control.

EXAMPLE: APPLYING A DMI CROSSOVER SYSTEM TO THE NASDAQ

OPEN AND EDIT A CHART

1. Enter the shortcut for the US Nasdaq ".NDX.X" in the Command Line.
2. Select the option **New Chart** from the drop-down menu.
3. Right-click into the empty area of the chart and select the **Properties** from the context menu.
4. Set the **History Length** to "2500" and press the ENTER key.
5. Double-click the time axis to display the full history.



DMI Crossover applied to the .NDX.X

SELECT THE STRATEGY

1. Click on the **Strategies** button in the Toolbox.
2. Search for the two "DMI Cross Over" systems and drag them into the chart.

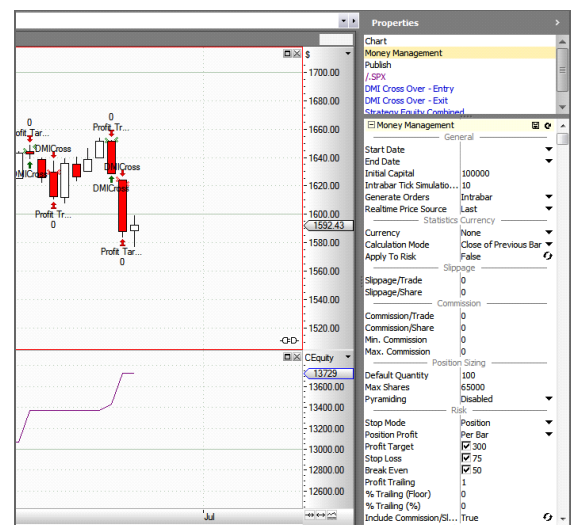
ENTER SIMPLE RULES FOR RISK CONTROL

1. In the upper area of the property inspector, click on Money Management.
2. Go to the area **Risk** in the properties.
3. Set **Stop Mode** to "Contract".
4. Select **Stop Loss** and enter "75".
5. Select **Break Even** and enter "50".
6. Select **Profit Target** and enter "300".

Now the strategy is configured.

APPLYING STRATEGIES TO SCANNER, WATCHLIST OR PORTFOLIO

You can apply strategies also to symbol tables such as the Scanner, Watchlist or Portfolio.



Detail: Money Management

- When creating a new Scanner, Watchlist or Portfolio, you can directly add strategies (and indicators).
- For existing tables, you can add strategies to the selected table by using one of the methods described above. A new column is added for the results.

DELETING STRATEGIES

DELETING A STRATEGY FROM THE CHART

There are two ways to delete a strategy from a chart:

- The strategy is displayed in the legend at the top left of the chart or subchart. Right-click on the entry to open the context menu and select **Delete**.
- Alternatively, select the entry in the legend and press the **DEL** key.

To delete a complete subchart of a strategy, click on the X button on the top right. Empty subcharts are closed automatically.

DELETING A STRATEGY FROM A SCANNER, WATCHLIST OR PORTFOLIO

- To delete a strategy from a table, right-click on the table header and select **Add/Remove Columns**.
- Alternatively, click on the **Columns** button and select **Add/Remove Columns** from the button menu. You can find more information in the chapters Scanner, Watchlist and Portfolio.

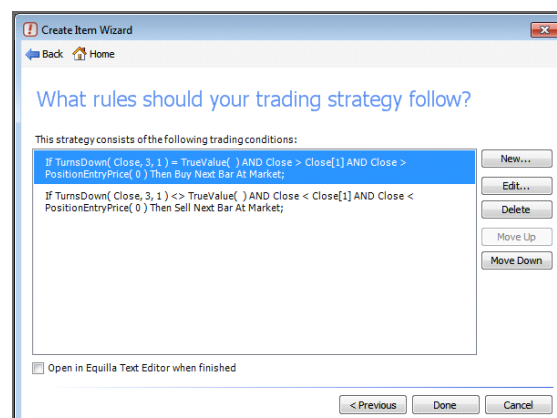
REMOVING A STRATEGY FROM TRADESIGNAL

To delete a strategy from the list in Tradesignal, right-click the strategy in the toolbox and select **Delete** from the context menu. A warning asks you to confirm the deletion.

CREATING A NEW STRATEGY

To create a new strategy, click on the link **New Strategy** in the *Related Tasks* area of the toolbox. A Wizard opens, offering two options:

- **Use the Equilla Formula Language to write the strategy** - Select this option if you want to write the strategy yourself. You can find more information in the chapter Writing Strategies.
- **Download an existing strategy from Tradesignal Online** - Select this option if you want to download a ready-made strategy from the website.
- **Use the Trading Strategy Wizard to create the strategy** - Select this option to use the Trading Strategy Wizard. It supports you in creating your own strategies without having to code in Equilla.



Strategy Wizard - start screen

DOWNLOADING A STRATEGY FROM TRADESIGNAL ONLINE

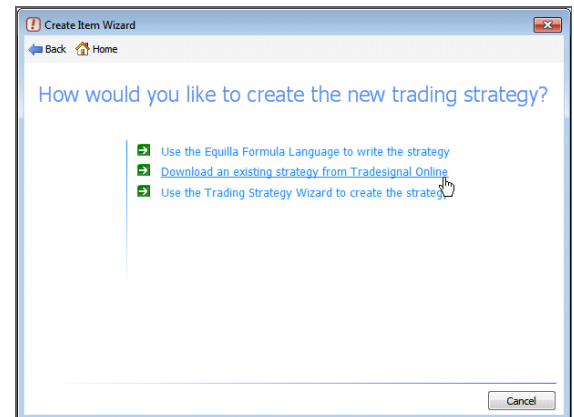
When you select the second option, the internal web browser opens and the entry "Handelssysteme" (strategies) in the Tradesignal lexicon appears.

Click on a strategy in the list. The entry for this strategy opens. At the bottom of the page, you will find one or more Equilla scripts (if available) and saving options:

- **Quelltext anzeigen** (show code) – Displays the Equilla code in the internal web browser. You can select and copy this code and, for example, paste it into the Equilla editor in the Tradesignal Standard Edition.
- **Skript in Tradesignal Web Edition importieren** (import into Tradesignal Web Edition) – Saves the script in the Tradesignal Web Edition. You can then find it in the list of all strategies in the web edition. To save the script in the web edition, you have to be logged into the Tradesignal Online website. In the case of the error message "Not enough space to import the script", you are either not logged in or there is not enough web space left to save the script.
- **Skript in Tradesignal Standard Edition importieren** (import into Tradesignal Standard Edition) – Saves the script in the Tradesignal Standard Edition. You can create the strategy in an existing package or create a new package.

In the case of a new package, the following procedure is recommended:

1. Click on the button **New package**.
2. Select the main package *Tradesignal Files*.
3. Click on **New Folder** and enter the name "Personal Strategies".
4. Confirm all settings by clicking on the **OK** buttons.



Strategy Wizard - download list

EDITING THE EQUILLA CODE OF A STRATEGY

- To edit the source code of a strategy, right-click the strategy in the chart legend and select **Edit Equilla Code** from the context menu.
- Alternatively, right-click the strategy in the toolbox and select **Edit** from the context menu.

The source code is opened in the Equilla Editor.

You can find more information in the Equilla chapters.

STRATEGY PROPERTIES

The strategy properties are available either from the context menu in the chart or the property inspector in the toolbox. Select the strategy from the list of elements at the top to open the properties.

Each strategy offers various parameters such as display units, period, uplevel, entry/exit methods etc. Enter all necessary parameters.

You can also edit the display properties of the trading system, e.g. the colors for long/short entries, price indicators, trade labels etc.

For some Tradesignal strategies it is possible to set an alert, which will be created as soon as a defined condition is met. Alerts will be displayed as a flashing icon in the system tray and as a popup window by default. Furthermore, you can find information about all alerts in the alert window. Added, cancelled or filled orders will be displayed on the *all alerts* tab.

CONTROLLING HOW AND WHEN ORDERS GET FILLED

In addition to the strategy properties, order execution is affected by the global trading properties located in the Money Management section. This section controls how orders get generated and filled, specifies how the slippage and commission that should be taken into account and defines the standard risk stop settings. The following settings directly affect how orders generated in a strategy get executed:

The **Realtime Price Source** property controls which prices are used to fill orders in real time. Tradesignal supports two modes: *bid/ask* and *last*.

- **Bid/Ask** - Tradesignal uses the best bid / best ask data provided by either the broker (if an order routing module has been connected) or the data provider. If neither the broker nor the data provider is delivering bid/ask prices, Tradesignal uses the last price instead (for example in the case of indices). Best bid and best ask prices are generated when a market participant wants to buy or sell a number of shares/contracts while a last price represents an actual trade that has been made. Therefore this option has the benefit of more accurate fills for your orders. The caveat is however that the results collected in real time will not match the results of back-testing which will always only use the last price.
- **Last** - Tradesignal uses the last price delivered by the data provider. This is the same price as the one used to back-test the strategy. A last tick represents a trade that has been made in the market. When using this mode, Tradesignal can only fill and order after a trade has been made, even if there was a matching offer/request in the market prior to the trade. This mode ensures that real time results are closer to back-test results.

The **Generate Orders** property specifies when orders can be generated by the running strategies.

- **Intrabar** - Orders can be generated with every intra-bar update (for every tick).
- **On Bar Close** - Orders can only be generated when the bar closes. This mode is especially useful when a signal might be invalidated before the bar closes, which typically happens when the signals are generated based on compressed chart types like Point and Figure, Renko, Three Line Break and Kagi (these chart types can all remove bars resulting in trades being executed at prices that may not be reflected in a bar when it is closed).

WRITING STRATEGIES

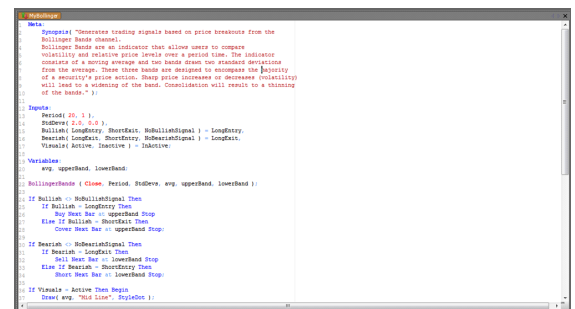
In Tradesignal, many strategies are available, either delivered with the software or offered for download from the website.

However, wishes may remain unfulfilled or you may have new ideas that need to be tested. You can always use the Trading Strategy Wizard to design a new strategy. You can also program new strategies (or edit available ones) in Tradesignal by using the Equilla language.

In the following you will find an example showing how to write a strategy using Equilla. It is only intended to give you a basic introduction. For details on Equilla, please refer to articles in the Equilla Basics category.

PROGRAMMING A BOLLINGER BAND - RSI SYSTEM

The strategy calculates the Relative Strength Index for an instrument. Based on these values, Bollinger Bands are then constructed. The crossings between the RSI and the Bollinger Bands are used as trading signals. In addition, Stop Loss and Trailing Stop are calculated. The idea behind this setup is to substitute the static extremes of the Relative Strength Index with more flexible zones that adapt to the movements of the indicator.



Writing a new strategy

CREATING THE STRATEGY

1. In the toolbox, click on the **Strategies** button.
2. In the *Related Tasks* area, select **New Strategy**.
3. In the next dialog, select the option **Use the Equilla Formula Language to write the strategy**.
4. Select a folder, e.g. "Tradesignal Files".
5. Type the name "Bollinger-RSI-Bands".
6. Click on **Done**.

Now you can enter the source code of the strategy in the Equilla Editor. The areas are described in the following sections. The complete code can be found at the end of this article.

ENTERING THE HEADER

In the header area, the input parameters of the used indicators and necessary variables are declared. In addition, we enter the meta information for the subchart creation.

```
//Creating a subchart
Meta:
    SubChart( true );

//Defining the input parameters for the indicators
Inputs:
```

```

PeriodAvg( 20 , 1 ),
PeriodStd( 10 , 1 ),
FactorStd( 2.0 , 0.0 ),
PeriodRSI( 14 , 1 ),
PeriodATR( 10 , 1 ),
FactorAtr( 1.5 , 0.0 ),
PeriodTrail( 20 , 1 ),
TradeMode( outsideIn, insideOut );

//Declaring the variables that we need for storing calculation results
Variables:
    avgValue, upperBand, lowerBand, rsiValue, stdValue, atrValue,
    stopValue, trailValue, activeStop;

```

PROGRAMMING THE CALCULATIONS FOR THE INDICATORS AND STOPS

In this part, the indicator calculations are entered. For most of them, we will use available Equilla Functions. However, a new calculation is written for the Bollinger Bands, as we want to use the RSI as their basis.

```

//Calculating the RSI
rsiValue = RSI( Close, PeriodRSI );
//Calculating the Bollinger Bands constituents (based on the RSI)
avgValue = Average( rsiValue, PeriodAvg );
stdValue = StdDeviation( rsiValue, PeriodStd );
//The upper band is calculated with the average and the added standard deviation
upperBand = avgValue + ( FactorStd * stdValue );
//The lower band is calculated with the average and the subtracted standard deviation
lowerBand = avgValue - ( FactorStd * stdValue );

//Calculating the Average True Range for the Stop Loss
atrValue = Average( TrueRange, PeriodAtr );
//Calculation the Average as Trailing Stop
trailValue = XAverage( Close, PeriodTrail );

```

DEFINING TRADING CONDITIONS AND PROGRAMMING ORDER GENERATION

In this part, the conditions for trading signals are defined and the orders are generated. Two modes are defined:

- The mode **outsideIn** is designed so that the RSI has to cut the Bollinger Bands from outside in for signal generation. This mode is preferable for correction movements.
- The mode **insideOut** is designed so that the RSI has to cut the Bollinger Bands from inside out, i.e. in trend direction. This mode is optimal for strong trends.

```
//Setting the trading conditions
//Signals are generated when the RSI crosses the bands from outside in
If TradeMode = 0 Then
    Begin
        //Sale when the RSI crosses the lower band from outside in.
        If rsiValue crosses over lowerBand Then
            Begin
                Buy("LongIn") Next Bar at Market;
                //Calculating the price for the Stop Loss
                stopValue = Close - ( FactorAtr * atrValue );
            End;

        //Short sale when the RSI crosses the upper band from outside in.
        If rsiValue crosses under upperBand Then
            Begin
                Short("ShortIn") Next Bar at Market;
                //Calculating the price for the Stop Loss
                stopValue = Close + ( FactorAtr * atrValue );
            End;
        End
    Else
        //Signals are generated when the RSI crosses the bands from inside out
        Begin
            //Sale when the RSI crosses the lower band from inside out.
            If rsiValue crosses over upperBand Then
                Begin
                    Buy("LongOut") Next Bar at Market;
                    //Calculating the price for the Stop Loss
                    stopValue = Close - ( FactorAtr * atrValue );
                End;

            //Short sale when the RSI crosses the upper band from inside out.
            If rsiValue crosses under lowerBand Then
                Begin
                    Short("ShortOut") Next Bar at Market;
                    //Calculating the price for the Stop Loss
                    stopValue = Close + ( FactorAtr * atrValue );
                End;
            End;
        End;
    End;
```

HANDLING OF THE STOP LOSS PRICES

For the Stop Loss, the Average True Range (ATR) is calculated. The value for the initial stop results from the ATR multiplied with a factor. As Trailing Stop, the Exponential Moving Average based on the instrument is calculated. If the price cuts this average inverse to the trade, the trade is closed. The program uses both stop values and compares which one is better positioned and more

protective of the trade. That value is finally output as a stop order for the next trading period.

```

/* Here the handling of stop values is done: For long positions: The stop loss is valid as
long as the trailing stop is below it. For short positions: The stop loss is valid as long
as the trailing stop is above it. */

//Recognizing long positions
If MarketPosition = MarketPositionLong Then
    Begin
        //If the stop loss is better positioned than the trailing stop
        //the stop loss is set
        If ( stopValue > trailValue ) Then
            Sell("Stop Loss") Next Bar at stopValue Stop;
        //If the trailing stop is better positioned than the stop loss
        //the trailing stop is set
        If ( trailValue > stopValue ) And ( trailValue < Close ) Then
            Sell("Trail") Next Bar at trailValue Stop;
    End;

//Recognizing short positions
If MarketPosition = MarketPositionShort Then
    Begin
        //If the stop loss is better positioned than the trailing stop
        //the stop loss is set
        If ( stopValue < trailValue ) Then
            Cover("Stop Loss") Next Bar at stopValue Stop;
        //If the trailing stop is better positioned than the stop loss
        //the trailing stop is set
        If ( trailValue < stopValue ) And ( trailValue > Close ) Then
            Cover("Trail") Next Bar at trailValue Stop;
    End;

```

DISPLAYING THE INDICATORS IN SUBCHARTS

This is the easiest part. Only the indicators that are used for signal generation are displayed here.

```

//Graphic display of indicators
DrawLine( rsiValue, "RSI", StyleSolid, 1, blue );
DrawLine( upperBand, "Upper Band", StyleSolid, 1, darkGreen );
DrawLine( lowerBand, "Lower Band", StyleSolid, 1, red );

```

APPLYING THE STRATEGY

Compile the code either by pressing **F7** or by clicking on the button **Compile Script** in the *Equilla Editor* group. The code is checked for errors, saved and then available in the list of strategies in the toolbox by the name "Bollinger-RSI-Bands".

You can now apply it to a chart it as described in Using Strategies, for example with drag and drop.

THE COMPLETE CODE

```
//Creating a subchart
Meta:
    SubChart( true );

//Defining the input parameters for the indicators
Inputs:
    PeriodAvg( 20 , 1 ),
    PeriodStd( 10 , 1 ),
    FactorStd( 2.0 , 0.0 ),
    PeriodRSI( 14 , 1 ),
    PeriodATR( 10, 1 ),
    FactorAtr( 1.5, 0.0 ),
    PeriodTrail( 20 , 1 ),
    TradeMode( outsideIn, insideOut );

//Declaring the variables that we need for storing calculation results
Variables:
    avgValue, upperBand, lowerBand, rsiValue, stdValue, atrValue,
    stopValue, trailValue, activeStop;

//Calculating the RSI
rsiValue = RSI( Close, PeriodRSI );
//Calculating the Bollinger Bands constituents (based on the RSI)
avgValue = Average( rsiValue, PeriodAvg );
stdValue = StdDeviation( rsiValue, PeriodStd );
//The upper band is calculated with the average and the added standard deviation
upperBand = avgValue + ( FactorStd * stdValue );
//The lower band is calculated with the average and the subtracted standard deviation
lowerBand = avgValue - ( FactorStd * stdValue );

//Calculating the Average True Range for the Stop Loss
atrValue = Average( TrueRange, PeriodAtr );
//Calculation the Average as Trailing Stop
trailValue = XAverage( Close, PeriodTrail );

//Setting the trading conditions
```

```
//Signals are generated when RSI crosses the bands from outside in
If TradeMode = 0 Then
    Begin
        //Sale when the RSI crosses the lower band from outside in.
        If rsiValue crosses over lowerBand Then
            Begin
                Buy("LongIn") Next Bar at Market;
                //Calculating the price for the Stop Loss
                stopValue = Close - ( FactorAtr * atrValue );
            End;

        //Short sale when the RSI crosses the upper band from outside in.
        If rsiValue crosses under upperBand Then
            Begin
                Short("ShortIn") Next Bar at Market;
                //Calculating the price for the Stop Loss
                stopValue = Close + ( FactorAtr * atrValue );
            End;
        End
    Else
        //Signals are generated when RSI crosses the bands from inside out
        Begin
            //Sale when the RSI crosses the upper band from inside out.
            If rsiValue crosses over upperBand Then
                Begin
                    Buy("LongOut") Next Bar at Market;
                    //Calculating the price for the Stop Loss
                    stopValue = Close - ( FactorAtr * atrValue );
                End;

            //Short sale when the RSI crosses the lower band from inside out.
            If rsiValue crosses under lowerBand Then
                Begin
                    Short("ShortOut") Next Bar at Market;
                    //Calculating the price for the Stop Loss
                    stopValue = Close + ( FactorAtr * atrValue );
                End;
            End;
        End;

        /* Here the handling of stop values is done: For long positions: The stop loss is valid as
        long as the trailing stop is below it. For short positions: The stop loss is valid as long
        as the trailing stop is above it. */

        //Recognizing long positions
        If MarketPosition = MarketPositionLong Then
            Begin
```

```

//If the stop loss is better positioned than the trailing stop
//the stop loss is set
If ( stopValue > trailValue ) Then
    Sell("Stop Loss") Next Bar at stopValue Stop;
//If the trailing stop is better positioned than the stop loss
//the trailing stop is set
If ( trailValue > stopValue ) And ( trailValue < Close ) Then
    Sell("Trail") Next Bar at trailValue Stop;
End;

//Recognizing short positions
If MarketPosition = MarketPositionShort Then
    Begin
        //If the stop loss is better positioned than the trailing stop
        //the stop loss is set
        If ( stopValue < trailValue ) Then
            Cover("Stop Loss") Next Bar at stopValue Stop;
        //If the trailing stop is better positioned than the stop loss
        //the trailing stop is set
        If ( trailValue < stopValue ) And ( trailValue > Close ) Then
            Cover("Trail") Next Bar at trailValue Stop;
    End;

//Graphic display of indicators
DrawLine( rsiValue, "RSI", StyleSolid, 1, blue );
DrawLine( upperBand, "Upper Band", StyleSolid, 1, darkGreen );
DrawLine( lowerBand, "Lower Band", StyleSolid, 1, red );

```

EXTENDING EQUILLA

C-API EXTENSIONS

Equilla, the formula language of Tradesignal, is a powerful way to describe your trading strategies. There are cases when you may need additional functionality which is not directly available in Equilla. To address this issue the language can be extended via Dynamic Link Libraries (DLLs) using the Equilla Extension Application Programmer's Interface (API).

WHAT IS AN EQUILLA EXTENSION

Equilla Extension libraries are DLLs containing one or more functions. These functions must have a specific signature in order to be compliant with Equilla.

HOW TO ACCESS AN EQUILLA EXTENSION FROM EQUILLA

To use an extension function from a script you have to first make sure the DLL is in the correct place on your system, and then you must add a reference to the function at the start of your Equilla script.

WHERE TO PUT THE EQUILLA EXTENSION

All Equilla Extension libraries have to be placed in the directory where Tradesignal is installed.

You can specify a different directory for Extension DLLs in the options under *File > Options > Folders > Equilla Paths*.

HOW TO USE EXTENSION FUNCTIONS

Assuming you have an Equilla Extension DLL called *MyEquillaExtension.dll* containing a function called *MyMovingAverage* that takes price and period parameters, you would first import the function using the following Equilla command:

```
Import ( "MyMovingAverage", "MyEquillaExtension.dll" );
```

You can then use the function as you would use a normal Equilla function.

```
DrawLine( MyMovingAverage( Close, 10 ), "Avg" );
```

Note: Each function from the DLL must have its own import statement.

HOW TO IMPORT A FUNCTION THAT HAS THE SAME NAME AS AN EXISTING EQUILLA FUNCTION

You can easily import a function and give it an alternate name as a third parameter to the Import command.

```
Import ( "MyMovingAverage", "MyEquillaExtension.dll", "RenamedMovingAverage" );
```

```
DrawLine( RenamedMovingAverage( Close, 10 ), "Avg" );
```

HOW TO WRITE AN EQUILLA EXTENSION IN MICROSOFT VISUAL C++ 2005

The Equilla Extension API is described in the *EquillaApi.h* file, which is located in the *Include* sub-directory of your Tradesignal installation. In order to create Equilla Extension libraries you have to add the directory to the *Include files* directory of Visual Studio under *Tools > Options > Projects and Solutions > VC++ Directories*.

Follow these steps to create a sample project:

- Create a new Visual C++ DLL project
 1. *File > New > Project... > Visual C++ > Win32 Console Application*
 2. Type a name for the new project for example "MyEquillaExtension"
 3. Press *OK*
 4. Select the *Application Settings* and set the *Application type* to *DLL*
 5. Press *Finish*
- Include the Equilla interface description and dependencies by adding the following lines to the top of the *MyEquillaExtension.cpp* file, below the *#include "stdafx.h"* line:

```
#include <wtypes.h>
#include <oleauto.h>
#include <EquillaApi.h>
```

- Add the following function to the bottom of the *MyEquillaExtension.cpp* file:
Note: All Equilla Extension functions must be declared with the following function prototype.

```
EQUILLA_API int MyMovingAverage( DWORD scriptID, int argc, EqVariable* variables[],
                                VARIANT* result )
{
    // check that the number of parameters passed to this function is two (Price and Period)
    if (argc != 2)
        return EQUILLA_ARGUMENT_ERROR;
    // extract the period value from the second parameter
    VARIANT vPeriod;
    VariantInit(&vPeriod);
    variables[1]->GetValue(&vPeriod); // read the latest value from the variable
    // ensure the period parameter is valid
    if (V_I4(&vPeriod) < 1)
        return EQUILLA_ARGUMENT_ERROR;
    // calculate the sum of all prices for the given period
    double sum = 0.0;
    VARIANT vPrice;
    VariantInit(&vPrice);
    for (int index = 0; index < V_I4(&vPeriod); ++index)
```

```

{
    variables[0]->GetValue(index, &vPrice); // read a historic value from the variable
    sum += V_R8(&vPrice);
    VariantClear(&vPrice); // release any allocated memory
}
// write the equilla result and set the type to a 8-byte floating point value
V_VT(result) = VT_R8;
V_R8(result) = sum / V_I4(&vPeriod);
// clean up and return successfully
VariantClear(&vPeriod);
return EQUILLA_SUCCESS;
}

```

- Compile the project and copy the DLL to the *Equilla Extension DLLs* directory
Note: Ensure you are compiling a 32-bit DLL otherwise it can not be used from Equilla!
- You are finished. You can now access this function from Equilla as described above.

EXTENSION FUNCTION ARGUMENTS AND RETURN VALUE

All Equilla Extension functions have the same prototype as shown above. The parameters are:

Parameter	Description
DWORD <i>scriptID</i>	A value that uniquely identifies the instance of an Equilla script calling this function
int <i>argc</i>	The number of parameters being passed into this function from the Equilla script
EqVariable* <i>variables[]</i>	An array of the parameters passed into this function from the Equilla script
VARIANT* <i>result</i>	A pointer to a variable receiving the result of the function that will be passed to the Equilla script

The return value is an integer that must be one of the following constants:

Return value	Description
EQUILLA_SUCCESS	The function has completed successfully
EQUILLA_ARGUMENT_ERROR	There is a problem with one or more of the parameters passed into the function
EQUILLA_GENERAL_ERROR	Any other error has occurred

WORKING WITH EQUILLA VARIABLES

The EqVariable object represents a variable that has been passed to this function from an Equilla script. You can use the object to access the latest value of the variable or a historic value (from a previous bar). All values accessed via the EqVariable object will be returned as VARIANTS.

Note: A VARIANT is generic variable that can hold data of various different types; for an overview of working with VARIANTS see below.

- To access the latest value of the variable:

```
VARIANT vValue;
VariantInit(&vValue);
variables[0]->GetValue(&vValue);
// convert the value into a string
VariantChangeType(&vValue, &vValue, VT_BSTR);
// include windows.h to use the MessageBoxW function
MessageBoxW(NULL, V_BSTR(&vValue), L"The value is", MB_OK);
VariantClear(&vValue);
```

- To access the value of the variable from the previous bar (assuming this is a series variable):

```
VARIANT vPreviousValue;
VariantInit(&vPreviousValue);
variables[0]->GetValue(1, &vPreviousValue);
// do something with the variable here
VariantClear(&vPreviousValue);
```

HOW TO DEBUG AN EQUILLA EXTENSION IN MICROSOFT VISUAL STUDIO 2005

Important: Before you start the debugger, ensure that Visual Studio is in native debugger mode. This can be enabled as followed:

1. Right-Click on the project in the Solution Explorer and select *Properties*
2. Click on the *Configuration Properties > Debugging* in the tree view on the left
3. Locate the *Debugger Type* property and change it's value to *Native Only*
4. Click *OK* to close the dialog

To start debugging your DLL:

1. Start Tradesignal and change the property *File > Tradesignal Options > Equilla > File Locations > Equilla Extension DLLs* to the build directory of your project (this will normally be in the sub folder called *Debug*). Click *OK*. Then shut down Tradesignal.
2. In Visual Studio ensure that this is the Debug build configuration and select *Debug > Start Debugging* from the menu.
3. A dialog will open asking you for the executable to run. Enter the location of the tse.exe file from the Tradesignal installation directory on your system and click *OK*.
4. The application will now start, you should use your extension function in a new script as described above.
5. Set breakpoints in visual studio and then apply your Equilla script to a Chart, Watchlist or Scanner. Execution will be automatically interrupted when one of your breakpoints is hit.

ADVANCED TOPICS

HOW DO I INITIALIZE AND CLEANUP GLOBAL DATA IN MY EQUILLA EXTENSION

If you want to write a library of functions that maintain some global state from call to call you will need to use global variables in your DLL. This can cause problems in determining when to initialize and free these global variables. To solve this problem, there are two optional functions that you can add to your extension DLL that will be called once when a script containing functions from your DLL is added to a chart, and once when it is removed from a chart. The following example illustrates how these functions are used:

Note: The prototypes and names of the *DllAttachScript()* and *DllDetachScript()* functions must be exactly the same as in the example.

```
#include "stdafx.h"
#include <wtypes.h>
#include <oleauto.h>
#include "EquillaApi.h"
#include <map>
BOOL APIENTRY DllMain(HMODULE hModule, DWORD dwReason, LPVOID lpReserved)
{
    return TRUE;
}
class CGlobalCounter
{
public:
    static CGlobalCounter& Instance()
    {
        // This is not thread-safe, included only for the brevity of the example
        static CGlobalCounter singleton;
        return singleton;
    }
    void Add(DWORD nScriptId)
    {
        EnterCriticalSection(&m_hCS);
        m_barCounts[nScriptId] = 0;
        LeaveCriticalSection(&m_hCS);
    }

    void Remove(DWORD nScriptId)
    {
        EnterCriticalSection(&m_hCS);
        m_barCounts.erase(nScriptId);
        LeaveCriticalSection(&m_hCS);
    }
    int Increment(DWORD nScriptId)
    {
        EnterCriticalSection(&m_hCS);
        int nResult = ++m_barCounts[nScriptId];
```

```

        LeaveCriticalSection(&m_hCS);
        return nResult;
    }
private:
    CGlobalCounter()
    {
        InitializeCriticalSection(&m_hCS);
    }
    CGlobalCounter(const CGlobalCounter&);
    CGlobalCounter& operator=(const CGlobalCounter&);
    std::map<DWORD, int> m_barCounts;
    CRITICAL_SECTION m_hCS;
};

EQUILLA_API void DllAttachScript(DWORD nScriptId)
{
    CGlobalCounter::Instance().Add(nScriptId);
}

EQUILLA_API void DllDetachScript(DWORD nScriptId)
{
    CGlobalCounter::Instance().Remove(nScriptId);
}

//call this function once on each bar to get the current bar count
EQUILLA_API int BarCount(DWORD nScriptID, int argc, EqVariable* variables[], VARIANT* result)
{
    V_VT(result) = VT_I4;
    V_I4(result) = CGlobalCounter::Instance().Increment(nScriptId);
    return EQUILLA_SUCCESS;
}

```

Note: Because this DLL can be used by multiple scripts instances at the same time it is important to separate the global data of each script in some way. In the example above we use a map that stores the bar count against a given scriptID. We recommend you use a similar approach in your own DLLs changing the int data type to whichever data you need to store.

WORKING WITH THE VARIANT DATA TYPE

A VARIANT is a variable that can hold any type of data, it maps quite closely to the types of variables that are used in Equilla. The trick to using variants in your Extension DLL is in correct initialization/de-initialization and in accessing the correctly typed data.

DECLARING AND RELEASING VARIANTS

Whenever you declare a variant in your code you should always initialize it in the following way:

```
VARIANT vPrice;
VariantInit(&vPrice);
```

Whenever you have finished with the variant (even in an error case) you should clear the variant in the following way:

```
VariantClear(&vPrice);
```

READING AND WRITING DATA

The following example demonstrates the important variant functions to use to read and write various data types from and to a variant:

```
// initialize the variant
VARIANT vValue;
VariantInit(&vValue);
// Write and read a double value (like a price) to a variant
double nPrice = 12.1;
V_VT(&vValue) = VT_R8; // set the underlying type of the variant
V_R8(&vValue) = nPrice;
double nNewPrice = V_R8(&vValue);
VariantClear(&vValue);
// Write and read an integer value (like a period) to a variant
int nPeriod = 14;
V_VT(&vValue) = VT_I4; // set the underlying type of the variant
V_I4(&vValue) = nPeriod;
int nNewPeriod = V_I4(&vValue);
VariantClear(&vValue);
// Write and read a Boolean value (a true/false value) to a variant
bool bValue = true;
V_VT(&vValue) = VT_BOOL; // set the underlying type of the variant
V_BOOL(&vValue) = bValue ? VARIANT_TRUE : VARIANT_FALSE;
bool bNewValue = V_BOOL(&vValue) != VARIANT_FALSE;
VariantClear(&vValue);
// Write and read a string value to a variant
LPCWSTR sValue = L"Text";
V_VT(&vValue) = VT_BSTR; // set the underlying type of the variant
V_BSTR(&vValue) = SysAllocString(sValue); // you have to create a BSTR from the input string
BSTR sNewValue = SysAllocString(V_BSTR(&vValue)); // SysFreeString needs to be called once sNew
VariantClear(&vValue);
// Write and read a date/time (OLE Datetime) to a variant
SYSTEMTIME stLocal = {0};
```

```
DATE dtOle = 0;
GetLocalTime(&stLocal);
SystemTimeToVariantTime(stLocal, &dtOle);
V_VT(&vValue) = VT_DATE;
V_DATE(&vValue) = dtOle;
DATE dtNewOle = V_DATE(&vValue);
VariantClear(&vValue);
```

CHECKING FUNCTION PARAMETERS ARE THE CORRECT TYPE

Due to the data types used by the Equilla engine, you should always check if the type of a given parameter is what you expect. If the type differs from what you expect you can use the *VariantChangeType* function to perform the conversion as the following sample demonstrates (this is the same moving average example as shown above but with this checking added):

```
EQUILLA_API int MyMovingAverage( DWORD scriptID, int argc, EqVariable* variables[],
                                VARIANT* result )
{
    // check that the number of parameters passed to this function is two (Price and Period)
    if (argc != 2)
        return EQUILLA_ARGUMENT_ERROR;
    // extract the period value from the second parameter
    VARIANT vPeriod;
    VariantInit(&vPeriod);
    variables[1]->GetValue(&vPeriod); // read the latest value from the variable
    // ensure the period parameter is an integer
    if (V_VT(&vPeriod) != VT_I4 && FAILED(VariantChangeType(&vPeriod, &vPeriod, VT_I4)))
    {
        VariantClear(&vPeriod);
        return EQUILLA_ARGUMENT_ERROR;
    }
    // ensure the period parameter is valid
    if (V_I4(&vPeriod) < 1)
        return EQUILLA_ARGUMENT_ERROR;
    // calculate the sum of all prices for the given period
    double sum = 0.0;
    VARIANT vPrice;
    VariantInit(&vPrice);
    for (int index = 0; index < V_I4(&vPeriod); ++index)
    {
        variables[0]->GetValue(index, &vPrice); // read a historic value from the variable
        // ensure the price is a double
        if (V_VT(&vPrice) != VT_R8 && FAILED(VariantChangeType(&vPrice, &vPrice, VT_R8)))
        {
            VariantClear(&vPeriod);
        }
    }
}
```

```

        VariantClear(&vPrice);
        return EQUILLA_ARGUMENT_ERROR;
    }
    sum += V_R8(&vPrice);
    VariantClear(&vPrice); // release any allocated memory
}
// write the equilla result and set the type to a 8-byte floating point value
V_VT(result) = VT_R8;
V_R8(result) = sum / V_I4(&vPeriod);
// clean up and return successfully
VariantClear(&vPeriod);
return EQUILLA_SUCCESS;
}

```

CONVERTING VARIANTS TO STRINGS

One of the most common conversion tasks is taking a variant (such as a price) and converting it into a string for output, this can be accomplished in the following way:

```

double nPrice = 12.1;
VARIANT vPrice;
VariantInit(&vPrice);
V_VT(&vPrice) = VT_R8;
V_R8(&vPrice) = nPrice;
// convert the value into a string
VariantChangeType(&vPrice, &vPrice, VT_BSTR);
// include windows.h to use the MessageBoxW function
MessageBoxW(NULL, V_BSTR(&vPrice), L"The value is", MB_OK);
VariantClear(&vPrice);

```

COM-API EXTENSIONS

The following article offers an introduction to COM-API Extensions for Equilla (**note:** available in Tradesignal 6.3 and above)

WHAT ARE COM-EXTENSION OBJECTS

Equilla COM-Extensions are COM-Objects that implement a special set of interfaces. Each COM-Object can provide one or more Extension objects by implementing one or more interfaces derived from the IEquillaExtension Interface. These objects are referenced from equilla by using the Prog-ID or the CLSID and an optional interface name. Extension COM-Objects need to be registered with the system so they can be instantiated via the Windows CoCreateInstance mechanism. For a detailed step-by-step guide to create COM-Extensions read the section How to write a COM Extension and check out the samples projects installed with Tradesignal.

HOW TO USE COM-EXTENSION OBJECTS

Equilla objects are created similar to Variables, by adding an *Objects*-Block to the script and declaring the objects you want to use. For every object within a script one extension object will be instantiated. If you use the same indicator multiple times separate extension objects will be created for every script instance.

```
Objects:
    myObject ("ProgID");

myObject.MyMethod();
```

For a detailed description of the Object syntax see the Equilla help.

Important Note: COM API Extension modules must be enabled in the options by selecting the setting at *File > Options > Advanced > Equilla Extension*.

LIFETIME OF AN EXTENSION OBJECT

All extension objects go through a common life cycle.

- Creation
- There are multiple reasons why an extension object can be created.
 1. By the equilla editor to provide auto completion support.
 2. By the compiler to validate that the script is correct.
 3. Because the compiled script is loaded (e.g. indicator added to a chart)
- If the extension object was loaded by the compile or editor no further actions will be performed and the object will be released shortly after. In case it was created by loading script the following actions will occur.
 - Attach Session (optional)
- After loading a compiled script the virtual machine will first attach a session object. This is an optional step that is only

performed if your extension implements the `ISupportSessions` Interface.

- Attach Host Object (`IEquillaExtension`)
 - This call on the `IEquillaExtension` Interface is done before the actual script evaluation starts. It supplies the extension object with a thread safe Host-Object that can be used to trigger reevaluation of the script and provides information about the executing context.
- Script Evaluation Phase
- While a script is evaluated bar-by-bar, methods will be called and signals will be passed to the extension object.
 1. Methods called by script evaluation

Once the script evaluation has started the virtual machine will call methods of your object as instructed by the script. These methods can contain multiple arguments of different type and a return value, as explained in [Extension Supported Parameter Types](#).
 2. Signals received (optional)

If your COM-Object implements any one of the Signal-Sink interfaces (`Extension Signals and Sinks`) signals will be passed to your extension after a bar is evaluated. `SymbolStateChanged` and interactive script alerts can be received at any time.
- Detach (`IEquillaExtension`)
- After script evaluation has stopped (e.g. indicator was removed from a chart) the `Detach()` method will be called on the `IEquillaExtension` Interface. Shortly after this the COM object will be released. All necessary clean up work should be performed at this point.

YOUR EXTENSION INTERFACE

The extension interface has to be derived from `IEquillaExtension` Interface and it should be the default interface of your COM-Object. If it is not the default interface the user will have to explicitly name it every time he wants to instantiate an equilla object.

Equilla extension interfaces requirements and limitations:

- Limited set of return and argument types (see [Extension Supported Parameter Types](#))
- Properties are not supported
- default values are not supported
- only 'dual' interfaces are supported

EXAMPLE PROJECTS

A few sample projects are shipped with Tradesignal. They are located in the install folder of Tradesignal.

- `ExcelOrderTracker (C#)`
- Demonstrates direct data transfer to Microsoft-Excel.
- `RSSExtension (C#)`
- Query of RSS 2.0 and Atom 1.0 data feeds, reacting on user interaction and display of custom user interface elements. This Extension is accompanied by the RSS-Indicator that can be found in the standard Tradesignal package.

- TextFileReader (C++)
- Reading and processing of text file from an equilla script.

HOW TO WRITE A COM EXTENSION

In this tutorial we will focus on using C# with Visual Studio 2008 (other versions of Visual Studio will work in much the same way). It is also relatively straight-forward to create extension objects in unmanaged C++ using ATL, and is preferable where raw performance is required. The TextFileReader sample project found in the installation directory of Tradesignal demonstrates how this is done.

OBJECTIVE

Create an Equilla extension object that can compute a Simple Moving Average and play a sound from the local hard disk.

CREATING A PROJECT FILE

The first task is to create an appropriate project:

- Start Visual Studio 2008 (as administrator when running Windows Vista or above)
- Select *File > New > Project... > Visual C# > Class Library*
- Enter a name for the Project (e.g. "SoundAverage") and click **OK**
- A project containing a single class called *Class1* should have been created

Once the project is created we must add a reference to the *EquillaExtension.dll*:

- Select *Project > Add Reference... > COM > Tradesignal EquillaExtension 1.0 Type Library* and click **OK**

DEFINING THE INTERFACE

The next task is to define what the interface for our new object will be and then to make it visible via COM:

Open the Class1.cs file if it is not already open and inside the *namespace SoundAverage* block add the following:

```
namespace SoundAverage
{
    public interface ISoundAverage
    {
        double SimpleAverage(EquillaExtension.IDoubleSeries prices, int length);
        void PlaySound(string path);
    }
    public class Class1
    {
    }
}
```

Next we must make the interface visible to COM, this requires us to indicate it is COM visible and to specify a unique class ID (a

GUID). Both of these can be set by specifying attributes as follows:

```
using System.Runtime.InteropServices;
namespace SoundAverage
{
    [ComVisible(true)]
    [Guid("E4AF0866-A1DB-462f-A304-4ED46EB7C2E6")]
    public interface ISoundAverage
    {
        double SimpleAverage(EquillaExtension.IDoubleSeries prices, int length);
        void PlaySound(string path);
    }
    public class Class1
    {
    }
}
```

Important: All classes and interfaces that we create must have a unique GUID, the tool under *Tools > Create GUID* is a simple way to create a new GUID.

Important: Once an interface is created and deployed it must **never** change, it is one of the ground rules of COM development. If you want to add more methods to a deployed interface you must create new interfaces.

ADDING HELP STRINGS TO THE INTERFACE

Extension objects are supported by the auto suggest and insight help features of the Equilla Editor. In order to display help messages we must adorn our interface and members with *Description* attributes:

```
using System.Runtime.InteropServices;
using System.ComponentModel;
namespace SoundAverage
{
    [ComVisible(true)]
    [Guid("E4AF0866-A1DB-462f-A304-4ED46EB7C2E6")]
    [Description("Calculates a moving average and plays sounds")]
    public interface ISoundAverage
    {
        [Description("Returns the result of a simple moving average")]
        double SimpleAverage(EquillaExtension.IDoubleSeries prices, int length);
        [Description("Asynchronously plays the .WAV file specified by the file name")]
        void PlaySound(string path);
    }
    public class Class1
    {
    }
```

```
}
}
```

IMPLEMENTING THE EXTENSION OBJECT

Now that we have our interface, we can create a class that uses it.

First of all we must implement our interface (we will ignore error handling for the sake of brevity, and because the Equilla compiler and runtime engine will display exceptions in a useful way if they occur):

```
public class Class1 : ISoundAverage
{
    double ISoundAverage.SimpleAverage(EquillaExtension.IDoubleSeries prices, int length)
    {
        double sum = 0;
        for (int i = 0; i < length; ++i)
            sum += prices.GetValue(i);
        return sum / length;
    }
    void ISoundAverage.PlaySound(string path)
    {
        System.Media.SoundPlayer player = new System.Media.SoundPlayer(path);
        player.Play();
    }
}
```

Next we must implement the *IEquillaExtension* interface, without this interface Tradesignal will not recognize the object:

```
public class Class1 : ISoundAverage, EquillaExtension.IEquillaExtension
{
    double ISoundAverage.SimpleAverage(EquillaExtension.IDoubleSeries prices, int length)
    {
        double sum = 0;
        for (int i = 0; i < length; ++i)
            sum += prices.GetValue(i);
        return sum / length;
    }
    void ISoundAverage.PlaySound(string path)
    {
        System.Media.SoundPlayer player = new System.Media.SoundPlayer(path);
        player.Play();
    }
    void EquillaExtension.IEquillaExtension.Attach(EquillaExtension.IEquillaHost Host)
    {

```

```

    }
    void EquillaExtension.IEquillaExtension.Detach()
    {
    }
}

```

We do not need to use the *IEquillaExtension* methods in this case.

We must next make the class visible to COM, indicate the default COM interface, and provide a suitable name (Prog ID) with which we will create the object from Equilla:

```

[ComVisible(true)]
[Guid("516024B3-0D76-4d5b-97AE-65F3A30E0C8A")]
[ComDefaultInterface(typeof(ISoundAverage))]
[ProgId("SoundAverage.Class1")]
public class Class1 : ISoundAverage, EquillaExtension.IEquillaExtension
{
    double ISoundAverage.SimpleAverage(EquillaExtension.IDoubleSeries prices, int length)
    {
        double sum = 0;
        for (int i = 0; i < length; ++i)
            sum += prices.GetValue(i);
        return sum / length;
    }
    void ISoundAverage.PlaySound(string path)
    {
        System.Media.SoundPlayer player = new System.Media.SoundPlayer(path);
        player.Play();
    }
    void EquillaExtension.IEquillaExtension.Attach(EquillaExtension.IEquillaHost Host)
    {
    }
    void EquillaExtension.IEquillaExtension.Detach()
    {
    }
}

```

Our COM-based extension object is almost ready, we just need to build it and register it.

- First indicate it should be registered automatically after it is built by checking the option: *Project > SoundAverage Properties... > Build > Output > Register for COM interop*
- Now, select *Build > Build Solution* to build the DLL.

Now we have a built and registered COM-based Equilla Extension Object which could now be deployed to other users workstations.

USING THE EXTENSION FROM EQUILLA

To use the object from Equilla we must first create a new indicator called *LoudSMA* after starting Tradesignal.

Next we must create an instance of our object as follows:

```
Object:
    soundAverage( "SoundAverage.Class1" );
```

Notice how we use the same ID string as we declared in the *ProgID* attribute when we created the object.

Finally we can call the functions on the *soundAverage* object to plot a SMA and make an audio alert based on a sound file when a cross over occurs (we assume the file exists at the specified location).

```
Inputs:
    Price( Close ),
    Length( 14, 1 ),
    AlertFile( "C:\alert.wav" );

Object:
    soundAverage( "SoundAverage.Class1" );

Variable:
    lsma;

lsma = soundAverage.SimpleAverage( Price, Length );

DrawLine( lsma, "LSMA" );

If Price Crosses Over lsma Then
    soundAverage.PlaySound( AlertFile );
```

HOW TO DEPLOY A COM EXTENSION

Once a COM Equilla extension has been created, it must be deployed to a target workstation. Deployment is fairly straight-forward, consisting of the following steps:

- Ensure prerequisites are installed (e.g. Tradesignal and correct version of C or .NET runtimes if Visual Studio 2008 is not used)
- Copy the Extension DLL
- Register the Extension DLL (either with *regsvr32.exe* for C++ DLLs or *regasm.exe* for .NET DLLs)
- Copy any indicator package created with Tradesignal
- Register the indicator package for auto discovery on Tradesignal start

These steps are best combined into a setup for ease of distribution. Packages such as InstallShield can handle all of the registration issues automatically.

DEPLOYING AN EXTENSION WITH A SETUP

The best way to distribute a library of indicators and/or strategies that include one or more Equilla extension DLLs is to use a setup.

First create the package in Tradesignal and copy the extension DLLs into the folder that contains the package contents using Windows Explorer.

Next, add this package folder including indicators, functions, strategies, manifest file and extension DLLs to a setup project (for example InstallShield). Already compiled indicators and strategies make it easier to use the deployed extension.

Mark any Extension DLLs as requiring registration:

- In InstallShield C++ DLLs should be marked as *Extract COM Information at Build*
- In InstallShield .NET DLLs should be marked as *Register for .NET COM interop*

Now instruct the setup to add a string value to the following registry key, with a unique name (GUID) and a value that contains the folder path on the target machine of the installed package:

`HKEY_LOCAL_MACHINE\Software\SystemSoft\TradeSignalEnterprise\Packages`

This will make Tradesignal automatically add the package when it is next started.

Finally, build and test the setup.

IEQUILLAEXTENSION INTERFACE

An extension object must implement the *IEquillaExtension* interface so that Tradesignal will recognize the object as a valid extension.

```
interface IEquillaExtension
{
    void Attach(IEquillaHost Host);
    void Detach();
}
```

When an object is first created the *Attach()* method will be called by Tradesignal with a Host object as the only parameter. The majority of initialization should be performed in the *Attach()* method rather than the objects constructor. The Host object is a thread-safe object that allows an extension module to communicate with the hosting chart or list.

When the Equilla script that declares the extension object is removed from a chart or list, the *Detach()* method will be called, allowing the extension object to preform clean-up. The majority of the clean-up of the object should be done in the *Detach()* method rather than the objects destructor.

IEQUILLAHOST INTERFACE

The interface provides access to methods on the chart or list that is hosting the extension object.

```
interface IEquillaHost
{
    void ScheduleEvaluation( bool Incremental );
    string TimeZone { get; }
    int OwnerHandle { get; }
}
```

The *IEquillaHost* interface is implemented by the *Host* object that is passed to each extension module shortly after creation. The *Host* object is thread-safe (neutral threaded) and may be passed between and used from various threads without restriction.

VOID SCHEDULEEVALUATION(BOOL INCREMENTAL)

Call this method to instruct the hosting chart or list to perform a new evaluation of the Equilla script that created the extension object. This method is non-blocking and the evaluation will occur in the near future in much the same way as if a tick had been received.

If *Incremental* is set to *true* only the current bar will be reevaluated, if *Incremental* is set to *false*, the entire price history and any dependencies will be reevaluated.

Please note: Performing frequent non-incremental evaluations will result in a *severe* reduction in system performance.

STRING TIMEZONE

Returns the current application timezone in Windows Timezone ID Format. This string can be used directly by the .NET *System.TimeZoneInfo.FindSystemTimeZoneById()* method.

```
System.TimeZoneInfo equillaTimeZone = System.TimeZoneInfo.FindSystemTimeZoneById( Host.TimeZone );
```

INT OWNERHANDLE

Returns the windows handle of an object that may be used as the parent of any windows that the extension object wants to show.

```
// Helper object to convert a windows ID to a IWin32Window object.
class WindowWrapper : IWin32Window
{
    public WindowWrapper( int handle )
    {
        Handle = new IntPtr( handle );
    }
    public IntPtr Handle { get; private set; }
```

```
}  
// Show a message box that is owned by the Tradesignal application  
System.Windows.Forms.MessageBox.Show( new WindowWrapper( Host.OwnerHandle ), "A message from a"
```

ISUPPORTSESSIONS INTERFACE

An extension object may implement the *ISupportSessions* interface in order to share a common *Session* object between all instances of the extension object used inside a single chart (or portfolio). Sessions provide a mechanism for multiple indicators within a chart to share a common resource, for example a connection to an external server.

```
interface ISupportSessions : IDispatch
{
    IEquillaExtensionSession CreateSessionObject();
    void AttachSession(IEquillaExtensionSession Session);
}
```

The first time an extension object is created that implements this interface, a chart will call the *CreateSessionObject()* factory method, which must return an object that implements the *IEquillaExtensionSession* interface. The newly created session will immediately be passed back to the extension object via a call to *AttachSession()*.

Following the creation of other extension objects of the same type within the same chart, the *AttachSession()* method will also be called with the original session passed as the only parameter.

The example project *RSSExtension* uses a session object to manage a pool of RSS feeds so that the same data does not need to be re-requested by each use of the RSS indicator within a chart.

The example project *ExcelOrderTracker* uses a session object to manage a single Excel workbook that all instruments within a chart will forward strategy data to.

IEQUILLAEXTENSIONSESSION INTERFACE

A session object may be created and shared between all instances of an extension object used within a single chart (or portfolio). Such a session object must implement the *IEquillaExtensionSession* interface.

```
interface IEquillaExtensionSession
{
    void Attach();
    void Detach();
}
```

An extension session object will only ever be created by an extension object that implements the *ISupportSessions* interface.

Shortly after the session object is returned from the *ISupportSessions.CreateSessionObject()* method, the *Attach()* method will be called. It is expected that the majority of the construction logic for the session object will be performed in this method instead of the object's constructor.

When the last extension object that uses a session is removed from the chart (or portfolio), the *Detach()* method will be called, so that the extension session can perform clean-up. It is expected that the majority of the destruction logic for the session object will be performed in this method instead of the object's destructor.

EQUILLACONSTANTS CLASS

The EquillaConstants object provides access to standard values used within Equilla scripts that may be needed within an extension object.

```
interface IEquillaConstants
{
    double InvalidValue { get; }
    DateTime InvalidDate { get; }
}
```

The *InvalidValue* constant can be used to test if a price parameter is set to the Equilla Invalid value.

The *InvalidDate* constant can be used to test if a date/time parameter is set to the Equilla Invalid value.

```
// Test if a price is valid
bool IsValidPrice( double price )
{
    EquillaConstants eqConsts = new EquillaConstantsClass();
    return price != eqConsts.InvalidValue;
}
```

EXTENSION SIGNALS AND SINKS

Extension objects can register interest in certain alerts and other events that may be generated by a chart (or portfolio) that are not directly related to an Equilla script evaluation. These *signals* are broken down into four main categories that can be independently hooked by the extension object: Equilla script alerts, tool alerts, strategy events and instrument events.

To register interest in a specific class of signals, the extension object must implements a *Sink* interface of the corresponding type:

```
interface IScriptSignal
{
    void ScriptSignalRaised( IScriptSignal Signal, bool MoreToFollow );
}
```

```
interface IStrategySignalSink
{
    void StrategySignalRaised( IStrategySignal Signal, bool MoreToFollow );
}
```

```
interface IToolSignalSink
{
    void ToolSignalRaised( IToolSignal Signal, bool MoreToFollow );
}
```

In each of the above cases, the implemented method will be called when a signal is raised, containing an interface to an object the contains details of that signal (see below), the *MoreToFollow* flag will indicate if more signals are pending, so that the signals may be queued and processed in a batch when *MoreToFollow* is set to *false* (this is of special importance with strategy orders).

```
interface ISymbolStateChangedSink
{
    SymbolStateChanged( string Symbol, string DisplayName, bool Active );
}
```

Implementation of the *ISymbolStateChangedSink* allows an extension object to respond when instruments transition from an active to an inactive state (inactive can mean the instrument is stale, or that the data connection has been lost).

ISCRIPTSIGNAL

Contains details about an Equilla alert that has occurred as the result of calling the Equilla *Alert()* function, or clicking on a chart hyperlink generated by *DrawText()* using the *signal* protocol.

Please refer to the sample project *RSSExtension* and the corresponding indicator *RSS* for how to hook and generate script alerts.

```
interface IScriptSignal
{
    string SourcePath { get; } // Unique workspace location of the event source
}
```

```
string DisplaySource { get; } // User visible location of the event source
string ScriptName { get; } // Name of the script that generated the event
string SignalName { get; } // Name of the event
string AlertText { get; } // Text specified in the Alert() function
string ParentSymbolName { get; } // Symbol of the script parent instrument
DateTime BarTimestamp { get; } // Chart timestamp of the bar on which the alert was generated
double BarPrice { get; } // Parent instrument price when the alert was generated
DateTime TimestampUTC { get; } // UTC Timestamp of the signal
string ParentDisplayName { get; } // Display name of the script parent instrument
}
```

A simple way to generate an alert would be to use the following Equilla code:

```
If IsLastBar Then
    Alert( "It is the last bar: " + CStr( CurrentBar ) );
```

To generate script alerts from a user clicking on a hyperlink in the chart, do the following:

```
If IsLastBar Then
    DrawText( High, "LinkTest", "Click 4 Alert", Default, Default, AlignTop,
        "signal:It is the last bar: " + CStr( CurrentBar ) );
```

ISTRATEGYSIGNAL

Contains details about a strategy event that has occurred. Strategy events are generated when a chart is running Equilla strategies that use the Buy, Sell, Short, Cover and ExitPosition functions to build and backtest a trading system.

Please refer to the sample project *ExcelOrderTracker* and the corresponding indicator *OrderTracker* for how to hook generated strategy alerts.

```
interface IStrategySignal
{
    EStrategySignalType SignalType { get; } // See below
    string SourcePath { get; } // Unique workspace location of the event source
    string DisplaySource { get; } // User visible location of the event source
    string ScriptName { get; } // Name of the script that generated the event
    string SignalName { get; } // Name of the event
    string OrderId { get; } // Unique ID of the order to which the event belongs
    EStrategySignalOrderType OrderType { get; } // See below
    double OrderPrice { get; } // Price of the order (valid for limit and stop orders)
    double OrderQuantity { get; } // Size of the order
    string ParentSymbolName { get; } // Symbol of the traded security
    DateTime BarTimestamp { get; } // Chart timestamp on which the order occurred
    double BarPrice { get; } // Chart price at which the order occurred
    DateTime TimestampUTC { get; } // UTC timestamp of the order
}
```

```

    string RelatedOrders { get; } // XML list of orders that will be generated if
    string ParentDisplayName { get; } // Display name of the traded security
}
enum EStrategySignalType
{
    eqSignalTypeLongEntryPlaced,
    eqSignalTypeLongExitPlaced,
    eqSignalTypeShortEntryPlaced,
    eqSignalTypeShortExitPlaced,
    eqSignalTypeOrderCancelled,
    eqSignalTypeOrderFilled,
    eqSignalTypeOrderModified,
    eqSignalTypeOrderModifiedTrailing,
}
enum EStrategySignalOrderType
{
    eqSignalTypeMarketLongEntry,
    eqSignalTypeMarketLongExit,
    eqSignalTypeMarketShortEntry,
    eqSignalTypeMarketShortExit,
    eqSignalTypeStopLongEntry,
    eqSignalTypeStopLongExit,
    eqSignalTypeStopShortEntry,
    eqSignalTypeStopShortExit,
    eqSignalTypeLimitLongEntry,
    eqSignalTypeLimitLongExit,
    eqSignalTypeLimitShortEntry,
    eqSignalTypeLimitShortExit
}

```

ITOOLSIGNAL

Contains details about a tool alert that has occurred.

```

interface IToolSignal
{
    EToolSignalType SignalType { get; } // See below
    string SourcePath { get; } // Unique workspace location of the event source
    string DisplaySource { get; } // User visible location of the event source
    string ScriptName { get; } // Name of the script that generated the event
    string SignalName { get; } // Signal text associated with the tool
    string ParentSymbolName { get; } // Symbol of the script parent instrument
    DateTime BarTimestamp { get; } // Chart timestamp of the bar on which the event occurred
    double BarPrice { get; } // Parent instrument price when the alert occurred
    EToolSignalToolType ToolType { get; } // See below
}

```

```

EToolSignalLine Line { get; } // See below
EToolSignalPriceDirection PriceDirection { get; } // See below
string ToolText { get; } // Text associated with the tool
DateTime TimestampUTC { get; } // UTC Timestamp of the signal
string ParentDisplayName { get; } // Display name of the script parent ins
}
enum EToolSignalType
{
    eqSignalTypeCrossed,
    eqSignalTypeBreakIn,
    eqSignalTypeBreakOut,
}
enum EToolSignalLine
{
    eqToolLine,
    eqToolUpperLine,
    eqToolMiddleLine,
    eqToolLowerLine,
    eqToolLineChannel,
    eqToolLineFibo1,
    eqToolLineFibo2,
    eqToolLineFibo3,
    eqToolLineFibo4,
    eqToolLineFibo5,
    eqToolLineFibo6,
    eqToolLineFibo7,
    eqToolLineFibo8,
    eqToolLineFibo9,
    eqToolLineFibo10,
    eqToolLineFibo11,
    eqToolLineFibo12,
    eqToolLineFibo13,
    eqToolLineFibo14,
    eqToolLineFibo15,
    eqToolLineFibo16,
};
enum EToolSignalToolType
{
    eqToolTypeTrendLine,
    eqToolTypeFiboRetracement,
    eqToolTypeStopLine,
    eqToolTypeTrendChannel,
    eqToolTypePitchfork,
    eqToolTypeRegressionChannel,
};

```

```
enum EToolSignalPriceDirection
{
    eqDirectionUpwards,
    eqDirectionDownwards,
    eqDirectionSideways,
};
```

EXTENSION SUPPORTED PARAMETER TYPES

Extension objects must declare an interface that may only contain methods (no properties or events). These methods may only accept as parameters the types listed below, and only as input. The methods may only return a value that is one of the *simple types* with the exception of *object*. A method may also have a return type of *void* - to return nothing at all.

SIMPLE TYPES

```
bool (IDL: VARIANT_BOOL)
int (IDL: long)
double (IDL: double)
string (IDL: BSTR)
DateTime (IDL: DATE)
object (IDL: VARIANT)
```

The *object* type should only be used when a function accepts different types for a specific parameter. The actual type of these parameters may change from call to call and can be of any of the supported types including extension objects (*IEquillaExtension*). Determining the actual type present is therefore a task for the extension object author. The *object* type is not the same as a parameter that accepts an extension object which should use the *IEquillaExtension* type (see below).

EXTENSION OBJECT TYPES

Any object that implements the *IEquillaExtension* interface can be passed to a method that has an *IEquillaExtension* parameter.

SERIES TYPES

Accepting a series type as input allows the extension object to access historic values of the parameter. There is no length value associated with the series types because the *BackBuffer* of the Equilla Script will be determined by how much historic data is accessed by the script and any extension objects. So for example if a script only needs 14 bars of data to process, this value should be provided to the extension object to ensure it does not blindly access historic data on a series variable and artificially elongate the *BackBuffer* (which could lead to severe performance problems).

In each of the below interfaces a single method exists, that when called will return the value at the supplied offset in the series variable. An offset of zero (0) implies the current value, one (1) the prior value and so on.

```
interface IDatetimeSeries
{
    DateTime GetDate( int Offset );
}
interface IDoubleSeries
```

```
{
    double GetValue( int Offset );
}
interface IIntegerSeries
{
    int GetValue( int Offset );
}
interface IStringSeries
{
    string GetValue( int Offset );
}
interface IBooleanSeries
{
    bool GetValue( int Offset );
}
interface IVariantSeries
{
    object GetValue( int Offset );
}
```

The *IVariantSeries* should only be used when a variable can contain a mixture of the other basic types (which is never a good thing to do). Determining the actual type present is therefore a task for the extension object author.

ARRAY TYPES

Array types accept as input Equilla Arrays (declared using the *Array*: block in an Equilla script). The extension API provides dedicated support for one- and two-dimensional arrays, and a generic version for n-dimensional arrays.

For each basic type supported by an array there is a corresponding series type with the same behavior as described for series types above. Array interfaces have a method to access a value at a given index (and offset in the case of series) and a property for determining the maximum index possible. Arrays are zero-based.

Arrays that hold *Variant* types should only be used when a variable can contain a mixture of the other basic types (which is never a good thing to do). Determining the actual type present is therefore a task for the extension object author.

ONE-DIMENSIONAL ARRAYS

```
interface IDatetimeArray
{
    DateTime GetDate(int Index);
    int MaxIndex { get; }
}
interface IDatetimeArraySeries
```

```
{
    DateTime GetDate(int Offset, int Index);
    int MaxIndex { get; }
}
interface IDoubleArray
{
    double GetValue(int Index);
    int MaxIndex { get; }
}
interface IDoubleArraySeries
{
    double GetValue(int Offset, int Index);
    int MaxIndex { get; }
}
interface IIntegerArray
{
    int GetValue(int Index);
    int MaxIndex { get; }
}
interface IIntegerArraySeries
{
    int GetValue(int Offset, int Index);
    int MaxIndex { get; }
}
interface IStringArray
{
    string GetValue(int Index);
    int MaxIndex { get; }
}
interface IStringArraySeries
{
    string GetValue(int Offset, int Index);
    int MaxIndex { get; }
}
interface IBooleanArray
{
    bool GetValue(int Index);
    int MaxIndex { get; }
}
interface IBooleanArraySeries
{
    bool GetValue(int Offset, int Index);
    int MaxIndex { get; }
}
interface IVariantArray
```

```
{
    object GetValue(int Index);
    int MaxIndex { get; }
}
interface IVariantArraySeries
{
    object GetValue(int Offset, int Index);
    int MaxIndex { get; }
}
```

TWO-DIMENSIONAL ARRAYS

```
interface IDatetimeArray2D
{
    DateTime GetDate(int Row, int Column);
    int GetMaxIndexOfDimension(int Dimension);
}
interface IDatetimeArray2DSeries
{
    DateTime GetDate(int Offset, int Row, int Column);
    int GetMaxIndexOfDimension(int Dimension);
}
interface IDoubleArray2D
{
    double GetValue(int Row, int Column);
    int GetMaxIndexOfDimension(int Dimension);
}
interface IDoubleArray2DSeries
{
    double GetValue(int Offset, int Row, int Column);
    int GetMaxIndexOfDimension(int Dimension);
}
interface IIntegerArray2D
{
    int GetValue(int Row, int Column);
    int GetMaxIndexOfDimension(int Dimension);
}
interface IIntegerArray2DSeries
{
    int GetValue(int Offset, int Row, int Column);
    int GetMaxIndexOfDimension(int Dimension);
}
interface IStringArray2D
{

```

```

    string GetValue(int Row, int Column);
    int GetMaxIndexOfDimension(int Dimension);
}
interface IStringArray2DSeries
{
    string GetValue(int Offset, int Row, int Column);
    int GetMaxIndexOfDimension(int Dimension);
}
interface IBooleanArray2D
{
    bool GetValue(int Row, int Column);
    int GetMaxIndexOfDimension(int Dimension);
}
interface IBooleanArray2DSeries
{
    bool GetValue(int Offset, int Row, int Column);
    int GetMaxIndexOfDimension(int Dimension);
}
interface IVariantArray2D
{
    object GetValue(int Row, int Column);
    int GetMaxIndexOfDimension(int Dimension);
}
interface IVariantArray2DSeries
{
    object GetValue(int Offset, int Row, int Column);
    int GetMaxIndexOfDimension(int Dimension);
}

```

N-DIMENSIONAL ARRAYS

```

interface IDatetimeArrayND
{
    DateTime GetDate(int[] Indices);
    int GetMaxIndexOfDimension(int Dimension);
    int Dimensions { get; }
}
interface IDatetimeArrayNDSeries
{
    DateTime GetDate(int Offset, int[] Indices);
    int GetMaxIndexOfDimension(int Dimension);
    int Dimensions { get; }
}
interface IDoubleArrayND

```

```
{
    double GetValue(int[] Indices);
    int GetMaxIndexOfDimension(int Dimension);
    int Dimensions { get; }
}

interface IDoubleArrayNDSeries
{
    double GetValue(int Offset, int[] Indices);
    int GetMaxIndexOfDimension(int Dimension);
    int Dimensions { get; }
}

interface IIntegerArrayND
{
    int GetValue(int[] Indices);
    int GetMaxIndexOfDimension(int Dimension);
    int Dimensions { get; }
}

interface IIntegerArrayNDSeries
{
    int GetValue(int Offset, int[] Indices);
    int GetMaxIndexOfDimension(int Dimension);
    int Dimensions { get; }
}

interface IStringArrayND
{
    string GetValue(int[] Indices);
    int GetMaxIndexOfDimension(int Dimension);
    int Dimensions { get; }
}

interface IStringArrayNDSeries
{
    string GetValue(int Offset, int[] Indices);
    int GetMaxIndexOfDimension(int Dimension);
    int Dimensions { get; }
}

interface IBooleanArrayND
{
    bool GetValue(int[] Indices);
    int GetMaxIndexOfDimension(int Dimension);
    int Dimensions { get; }
}

interface IBooleanArrayNDSeries
{
    bool GetValue(int Offset, int[] Indices);
    int GetMaxIndexOfDimension(int Dimension);
}
```

```

    int Dimensions { get; }
}
interface IVariantArrayND
{
    object GetValue(int[] Indices);
    int GetMaxIndexOfDimension(int Dimension);
    int Dimensions { get; }
}
interface IVariantArrayNDSeries
{
    object GetValue(int Offset, int[] Indices);
    int GetMaxIndexOfDimension(int Dimension);
    int Dimensions { get; }
}

```

EXAMPLE: PASSING AN ARRAY TO AN EXTENSION DLL

C# interface and code for the extension DLL (all other series and array types are used in much the same way):

```

public interface ISoundAverage
{
    [Description("Asynchronously plays the .WAV files specified by the array of file names")]
    void PlaySounds(IStringArray paths);
}
[ProgId("SoundAverage.Class1")]
public class Class1 : ISoundAverage, EquillaExtension.IEquillaExtension
{
    void ISoundAverage.PlaySounds(IStringArray paths)
    {
        for(int i = 1; i < paths.MaxIndex; ++i)
        {
            System.Media.SoundPlayer player = new System.Media.SoundPlayer(paths.GetValue(i));
            player.Play();
        }
    }
    void EquillaExtension.IEquillaExtension.Attach(EquillaExtension.IEquillaHost Host)
    {
    }
    void EquillaExtension.IEquillaExtension.Detach()
    {
    }
}

```

Usage from Equilla:

Inputs:

```
Price( Close ),
Length( 14, 1 );
```

Array:

```
pathArray[] ();
```

Object:

```
soundAverage( "SoundAverage.Class1" );
```

```
If CurrentBar = 1 Then
```

```
    FillArray( pathArray, "c:\file1.wav", "c:\file2.wav" );
```

```
If Price Crosses Over Average( Price, Length ) Then
```

```
    soundAverage.PlaySound( paths );
```

DATA MANAGEMENT

DATA FEEDS

Here you can find a list of data providers supported by the Tradesignal family.

THOMSON REUTERS

Available for **Tradesignal**. Further information for existing customers using a Thomson Reuters data feed can be found in the **Thomson Reuters Data** section. A list of all mayor symbol codes you can find here: **Reuters Symbolists**. Reuters users interested in testing Tradesignal can find more information online at www.tradesignal.com.

BLOOMBERG

Available for **Tradesignal**. Further information for existing customers using a Bloomberg data feed can be found in the **Bloomberg Data** section. A list of all mayor symbol codes you can find here: **Bloomberg Symbolists**. Bloomberg users interested in testing Tradesignal can find more information online at www.tradesignal.com.

TRAYPORT GLOBALVISION

Available for **Tradesignal**. Trayport users interested in testing Tradesignal can find more information online at www.tradesignal.com.

TELETRADER

Available for non-professional users as part of **Tradesignal Online Terminal**, more information and a free trial for this product can be found online at terminal.tradesignal.com.

Available for professional users with **Tradesignal**. Further information for existing customers using a TeleTrader data feed can be found in the **TeleTrader Data** section. TeleTrader users interested in testing tradesignal enterprise edition can find more information online at www.tradesignal.com.

MISSING A DATA PROVIDER?

You would like to see the list of data providers expanded? No problem - contact us at sales@tradesignal.com to discuss the possibility of connecting another data provider to Tradesignal.

SESSIONS, HOLIDAYS AND PROPERTIES

Information on basic data properties, sessions, time zones and holidays are usually offered by the data provider together with the quotations. However, the data may contain errors - for whatever reason - or the data may cause unwanted side effects in your trading system or the chart display. For cases like these, Tradesignal offers you the possibility to add missing information or even overwrite the delivered data. Use the editing for:

- Adjusting currency data to your working hours
- Adding the right trading times (sessions) to quotations
- Adding time zone information for quotations
- Adding missing holidays or other days without trading

MODIFY SESSIONS AND PROPERTIES WINDOW

The window for editing the sessions, holidays and properties can be called up for an instrument (a symbol). This can be done in a Chart, or from a selected row in the Scanner, Watchlist or Portfolio. You can open the window in two ways:

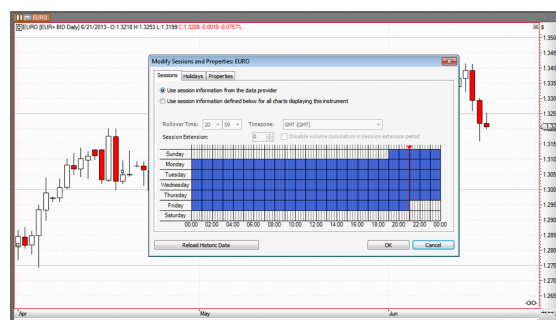
- Click on the **Sessions** button in the *Home > Data* group on the Toolbar.
- Alternatively, right-click on the chart legend or the selected row and choose the option **Show Session Manager** in the context menu.

The window opens. The name of the currently edited instrument is displayed in the window header.

DEFINING SESSIONS

On the tab *Sessions* you can set up which times should be used for trading. You have two options:

- **Use session information from the data provider** - Select this option if you want to use the information delivered by the provider. E.g., the trading hours of the NYSE are 9:30 - 16:00 local time.
- **Use session information defined below for all charts displaying this instrument** - Select this option to set up your own trading times for a symbol.



Session Manager, Tab Sessions

The trading hours are displayed as blue areas in the session display. Move the mouse over the cells to see which time range they represent.

You can select the trading hours as follows (repeating the actions removes the selection again):

- Click on the name of each weekday to add (or remove) complete days from the selection.
- Keep the mouse button pressed and move the mouse over the upper edge of the session display to select a time block, e.g. 08:00 to 15:00.
- Click on a column header to select a cell column.
- To define session times to the minute, right-click a cell and select another start time from the list.

- To select a continuous region, keep the mouse button pressed while moving over the cells in the session display.

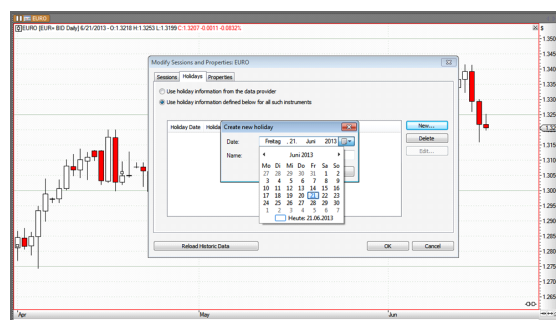
You can select a **Time Zone** for which the quotations are displayed. This setting is important when you are using external quotations from distant exchanges and the original data does not include the time information. You can also set the time zone when defining external data sources (see the chapter Data Sources); however, in this case the settings affect all data in the defined folder, even when they do not belong to the same time zone.

When you want to display trading information after the official end of the session without new bars/candles being generated, select the option **Session Extension**. Enter a number of bars or click on the small arrows to enter the numbers. The information for high, close, low and volume are added to the last bar. When you select the additional option **Disable volume cumulation in session extension period**, only the prices, not the volume are added to the last bar.

DEFINE HOLIDAYS

On the tab *Holidays* you can define which holiday options should be used. There are two options:

- **Use holiday information from the data provider** - Select this option to use information delivered by the provider. (Note: not every Dataprovider delivers holiday information)
- **Use holiday information defined below for all such instruments** - Select this option to enter holidays. You can define any day as "holiday", i.e. day without trading activity, for example bridging days.



Session Manager, Tab Holidays

To add a new holiday, click on **New**. A dialog opens. To enter the date, click on the arrow button to open a calendar in which you can select a date. Then enter a name for the holiday.

To edit a holiday, select it in the list and click on **Edit**. A dialog opens in which you can edit the date and/or the name.

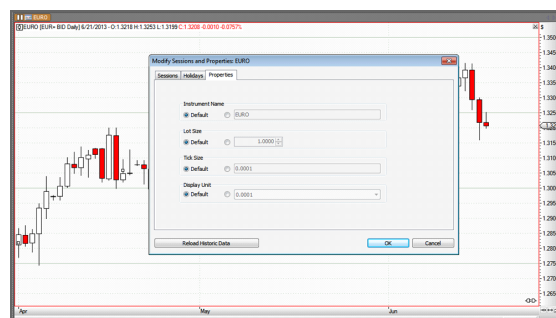
To delete a holiday, select it in the list and click on **Delete**.

EDIT SECURITY PROPERTIES

On the tab *Properties* you can change four settings for the instrument data. This is especially useful when the external data does not include this information. As default, the *Standard* settings are selected.

Instrument Name - Here you can change the name of the symbol as displayed in the chart.

Lot Size - The Lot Size is the smallest number of pieces traded, for example 100 pieces for a security. For futures, it is usually 1 piece. To simulate a leverage effect that is caused by multipliers, you can edit the lot size. For the Dax Future that would be 25 Euro, for the Bund Future 1000 Euro.



Session Manager, Tab Properties

Tick Size - There you can change the smallest possible price movement of a security. For stocks and some indices, it is 0.01 Cent. For some futures it is 0.5 points, for other futures 0.01 ticks or for currencies 0.0001 pips.

Display Unit - Here you can change the smallest unit of the value scale.

IMPORTING EXTERNAL DATA

Today, all charting software is connected to the internet and thus also to a data provider. This data is called "primary data feed" in Tradesignal. For intraday trading, it is important to have up-to-date and self-refreshing data.

For some trading methods or for developing trading systems and tests, however, it is important to have long histories and a high time resolution. These histories are often expensive and can only be purchased as monthly subscriptions. A less expensive and more individual solution is the inclusion of historic data of files saved on hard drives, removable storage or web servers. For such quotation data, many free and pay sources are available. This allows the users to retrieve and pay only for the histories they need for their analysis.

Offline quotations from external sources can be used for:

- Extending the price histories of the online data provider
- Inserting instruments not offered by the online data provider
- Inserting data other than quotations (business or sentiment indicators, statistics)

External data is treated the same as data from the online provider. You can apply all functions like Chart Analysis, Indicators, Strategies, the Portfolio and the Scanner.

In the following, you can find information about the import of external data and the necessary structure of the file name and the data of such quotation files.

IMPORTING EXTERNAL DATA IN TRADESIGNAL

EDITING THE DATA FEEDS

You can open the settings for the data feed in two ways:

- Double-click the connection icon in the top-right corner of the Tradesignal window.
- Alternatively, open the file menu, select the **Options** entry and click on **Datafeeds**.

The area for data connection appears in the right side of the window. At the top you can find settings for the data connection to online providers, for example "Tradesignal Online Terminal". Below you can find the settings for the "additional data sources" that are available offline.



Entry Datafeed in the Tradesignal Options

DATAFEED (ONLINE DATA)

Here you can set up the primary data feed.

- **Provider** - Select your online provider here.
- **Settings** - Here you can see the settings of your connection. Different data feeds have different settings. To edit an entry,

click on the value to the right, for example "300" for the parameter "Keep Alive Interval".

Click on the **Show more options** link to show less critical configuration options.

When you make changes to the data feed, a link will appear offering the chance to reconnect now. Alternatively, clicking on the connection button at the top right above the toolbar will show a menu with a reconnect option.

CSV DATA SOURCES (OFFLINE DATA)

Here you can edit additional data sources. Depending on the settings, this data is used in addition to or as a substitution for online data.

Any available path can be used as data source. You can define folders on local hard drives, external hard drives or network drives as additional data sources. It is also possible to use web or XML servers.

The list of data sources determines the order in which Tradesignal searches these folders. When you use a symbol in a chart, this is important if you have more than one folder for the same symbol. In this case, the order determines which data is displayed.

- **New** - Create a new data source. A dialog opens, see next section.
- **Edit** - Edit the selected data source.
- **Delete** - Delete the selected data source. Only the source reference is deleted, not the data on the drives.
- **Up/Down** - Use these buttons to change the sequence of folders in the list.

ADDING AND EDITING ADDITIONAL DATA SOURCES

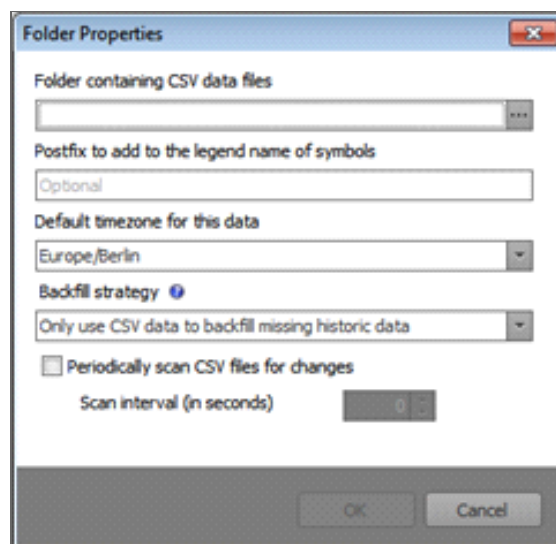
Set up the type of the additional data source:

- **Add a folder containing CSV data** - To configure the folder, click on **Browse**. Select the folder in which the imported files are located. This may be any folder available on your computer or via intranet. You can also create a new folder for future import. The folder is also called "top-Up service", as it works like a data provider.

Please note that external quotation files can only be used when they are located in the configured folder and adhere to both the Naming Conventions for Files and the Data Formats.

- **Add a TXML server** - Select this option if the external data is to be found on a web or XML server. Enter a URL.

Select the **Timezone** from the list (see above).



Window view "Location of Data Source"

You can enter a **Text that is added to the legend name of an instrument**. The default string is "top-up". You can enter any other string or delete the entry and leave the field empty. The additional label on the offline data is intended to help you distinguish them from online data. This is especially useful if both data sets have the same shortcut. Additional label won't be visible when extending the price histories of the online data.

COMBINING ONLINE AND OFFLINE DATA

If symbol data is available online and offline (top-up), you can configure how Tradesignal should proceed:

Only use data to back-fill missing historic data (default setting) - Usually, the online data provider offers all instruments needed. However, the available history is often too short or no intraday data is available. If you want to fill the missing history by data of offline sources, the files have to be saved in the folder configured above and the data shortcut has to be the same as for the instrument of the online provider. Only if both conditions are met, Tradesignal can concatenate the data. The program attempts to fill the history into the past. The additional data is inserted at the point where the history of the online data provider ends.

Always overwrite historic data - This setting is only sensible if you have enough offline data for this instrument, so that most or all of the online data is unnecessary. Alternatively, you could also enter a new instrument name so that no collision or concatenation could take place. However, if you want to use a general shortcut for the data without using the online data, this is the right option.

CONDITIONS FOR THE CORRECT DISPLAY OF TRADING PERIODS

For Tradesignal to recognize the time resolution (period) of your data, certain conditions have to be met for the File Names and the Data Format. In cases of display problems or certain trading periods missing in the chart although the data is available, the error can usually be tracked down to one of these conditions.

EXAMPLE: CONFIGURE AN ADDITIONAL DATA SOURCE IN TRADESIGNAL

STEP 1: OPENING THE CONNECTION SETTINGS IN TRADESIGNAL

1. Open the file menu to the left of the command line.
2. Click on the menu point **Options**.
3. Click on the tab **Datafeeds**.

STEP 2: DEFINE A FILE FOLDER AS DATA SOURCE

1. Click on **Add a folder containing CSV data** to configure a new connection.
2. Click on **Browse** to open the folder list.
3. Browse to the folder "Tradesignal Files", select it and click on **New Folder** to create a new folder.
4. Enter the name "Ascii Data" and click on **Create**.
5. Confirm with **OK**.
6. As **Time Zone**, select "Europe/Berlin".
7. Confirm with **OK**.

STEP 3: IMPORT AND SAVE FILE

Save the file Sample02-1d.txt in the folder described above (right-click the link and "save as target").

If you have read the article about File Names, you can see that the shortcut for the instrument is *Sample02* and that the data has a resolution of one day. Keep this instrument name for now. You can always change it later, if desired.

STEP 4: OPEN CHART

1. Enter the shortcut "Sample02" in the command line.
2. Select the option **New Chart** from the drop-down menu.

The chart with the "Sample02" data is opened.

FILE NAMES FOR QUOTATION DATA

File names have an important function - ideally, they tell the user what kind of data is included and how it is to be displayed. The operating system automatically opens the program assigned to the file extension.

In the case of quotation data like in our case, the file name has another important function. It shows us and the software to which instrument the data belongs, which trading period is used and more. If you have a problem displaying data, this is probably caused by an error either in the file name or in the Data Format.

Please note where the information in a filename is used and if it is mandatory or optional.

FILE NAMES

The file name can/must include the following information:

- Shortcut (mandatory)
- Period (mandatory)
- Field Identifier (optional)

File names can be constructed in the following ways:

- Shortcut.FileExtension
- Shortcut-Period.FileExtension
- Shortcut-FID-Period.FileExtension

As a separator between the parameters, the minus sign is used. If you want to use a shortcut that already includes a minus sign, substitute it with a comma or semicolon. Valid file extensions recognized by Tradesignal are .csv, .asc and .txt.

SHORTCUT

If you want to be able to use the file name in the Command Line of Tradesignal, e.g. to open a chart with the data, the shortcut has to follow a certain convention. The shortcut is also important if you want to combine data from your online provider with offline data.

Notation: If you want to use a shortcut independent of your online provider, you may freely choose the filename. However, if you want to combine data from your online provider with offline data (for example to extend the history or to add intraday data for an

instrument), the shortcut of the offline data has to be identical to the one used by the online provider. Otherwise, the external offline data will not be assigned correctly.

For shortcuts with a leading point (e.g. many indices), a trick has to be used since Windows does not allow filenames with a leading point. Just add a plus sign before the point.

Example for an instrument with leading point: +.gdaxi-5m.csv

PERIOD

The period is optional in a file name. If no period is given, Tradesignal automatically displays the data in daily intervals, even if intraday data is available in the file. The software can display the data correctly only if the period parameter is used. If you receive an error when changing the period settings of your external data in the Chart or Scanner, it is very likely that this error results from a missing or invalid period value.

Notation:

Period	Short code	Example
Tick Data	tick, 0, t, tick, ticks, 1t	aaa-tick.csv
1 Minute	60, 1m, m, min, minute, minutes, minuten	pumg.de-60.csv
5 Minutes	300, 5m, 5min, 5mins, 5minute, 5minuten, 5minutes	adsg.de-5m.csv
30 Minutes	1800, 30m, 30min, 30mins, 30minute, 30minuten, 30minutes	depfa-30min.csv
1 Hour	3600, 1h, h, hour, hourly, 1hour, stunde, stunden	eurusd-hour.csv
1 Day	86400, 1d, d, day, daily, tag	+.rut-1d.txt
1 Week	604800, 1w, w, week, weekly, woche	+.gdaxi-w.csv
1 Month	2592000, month, mothly, monat	dowjones-monthly.csv

FIELD IDENTIFIER

The Field Identifier is only mandatory if instead of the normal quotations (open, high, low and close) the bid, ask, or last prices are given in the file. These prices can only be displayed with the correct identifier. The following identifiers are valid:

- last
- bid
- ask

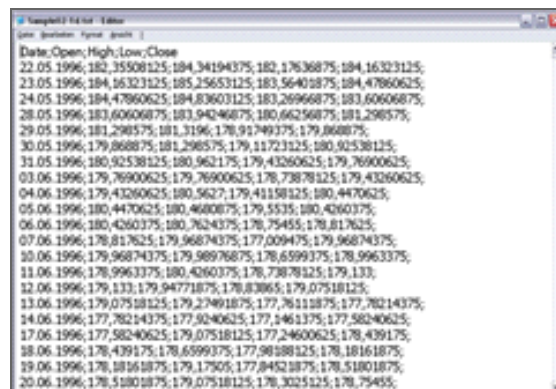
Example for a file with bid prices: +.gdaxi-bid-5min.csv

Example for a file with ask prices: +.gdaxi-ask-5min.csv

DATA FORMAT OF EXTERNAL QUOTATIONS

In files of quotation data, the data format (i.e. the sequence of elements such as date, time, prices and volume) defines the data display and whether the program can handle it at all. Since there are many possible combinations, import problems may be caused by data in the wrong format. Often, a simple wrong separator or a non-supported date format can cause havoc with the data.

When you need to edit the data format, you can do this with a text editor. For exchanging complete columns, however, usually programs like Excel or Access are necessary.



View of an ASCII file in a text editor

A file with quotation data has the following components:

- Header - recommended but optional
- Data columns separated by separators (by default semicolon or comma)

Tradesignal recognizes the following column types:

- Date
- Time - optional
- Open - optional
- High - optional
- Low - optional
- Close
- Volume - optional
- Open Interest - optional

For the display of intraday data, time and date stamps are necessary. Otherwise, the data is filled in beginning with the current trading day, which may not always give correct results. If you want to add information to the history of your online data provider, make sure that your file has the same data elements as the file from your provider.

HEADER

The first segment of the file should be the header. It contains information about the data columns, their order and the used separators. The header is optional; when no header is given, Tradesignal attempts to interpret and display the segments. However, this may cause errors.

In the header, information about the columns is given in the way of identifiers. The identifiers have to be separated by the same separator as the data columns. For each column type, several identifiers are possible, as you can see in the following list:

Notation

- Date - d, date, datetime (when date and time are combined)
- time - t, time
- Open - o, op, open
- High - h, hi, high
- Low - l, lo, low
- Close - c, cl, close

- Volume - v, vo, vol, volume
- OpenInterest - oi, openint, ointerest, openinterest

Example Header: Date;Time;Open;High;Low;Close;Volume

If no header is given, Tradesignal uses the following masks to analyze the column grouping:

- d,o,h,l,c
- d,o,h,l,c,v
- d,o,h,l,c,v,oi
- d,c
- d,c,v
- d,c,v,oi

Some data sources, for example Internet offers, contain data in additional columns. Since Tradesignal cannot display those, the data has to be skipped. For this, the word "Skip" can be used in the header. In the following header example, the last column is skipped.

Example for a file with an additional column that is skipped:

```
Date,Open,High,Low,Close,Volume,Adj. Close*09.11.05,10539.24,10637.78,10466.24,10546.21,22144600,10546.21
```

results in:

```
Date,Open,High,Low,Close,Volume,Skip09.11.05,10539.24,10637.78,10466.24,10546.21,22144600,10546.21
```

DATA AND TIME FORMATS

For date and time, many different formats exist.

DATE FORMAT

- European Format - Example: 01.04.2006
- English (US) Format - Example: 04/01/06
- Standard ISO Format - Example: 1998-12-22T01:02:03:987

When the European format is recognized, a comma has to be used as decimal separator. In the case of an English format, a decimal point has to be used as decimal separator.

Additional differences may exist in the day, month, year sequence, especially in the case of UK/US formats. See the example below for combinations recognized by Tradesignal.

TIME STAMP

No flexibility is given regarding the time stamp format. The separator has always to be a colon. If necessary, the separators of an external data file have to be exchanged to a colon in a text editor.

Components of the Time Stamp

- hour
- minute
- second
- one-tenth of a second (for tick data)

For English time zones, "AM" or "PM" can be used if the hours are given in a 12 hour interval.

EXAMPLES FOR VALID DATE AND TIME STAMPS IN TRADESIGNAL

ENGLISH FORMATS

- 1998/1/11 11:05:00
- 5/17/1998 11:05:00 AM
- 5/17/1998 8:05:00 PM
- 2005/10/25 12:25:11 PM
- 2005/10/25 11:05:00:124 AM
- 5-17-1998
- 1998-12-22
- 1998-12-22 01:02 PM
- 98-12-22

EUROPEAN FORMATS

- 04.12.98
- 04.12.1998
- 04.12.1998 12:23
- 04.12.1998 12:23:12
- 04.12.1998 12:23:14:122
- 04.12.1998 12:23:14.122

ISO FORMATS

- 1998-12-22T01:02:03
- 1998-12-22T01:02:03:987

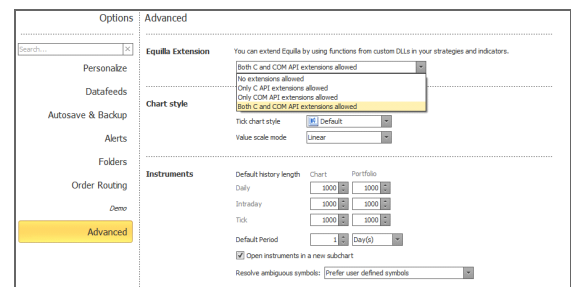
TRADESIGNAL DATA IN EXCEL

Extending Equilla via COM-API Extensions allows you to get periodically updated data from Tradesignal into your Excel spreadsheets.

In Tradesignal 7, we've added a COM-API extension *LiveExport* and indicator of the same name to help you accomplish this. The extension enables the indicator to write price data into an Access database file which can be imported into an Excel worksheet. To get periodic updates to the data displayed, you can specify how often Excel updates from the database.

WRITING DATA FROM A CHART TO AN ACCESS DATABASE

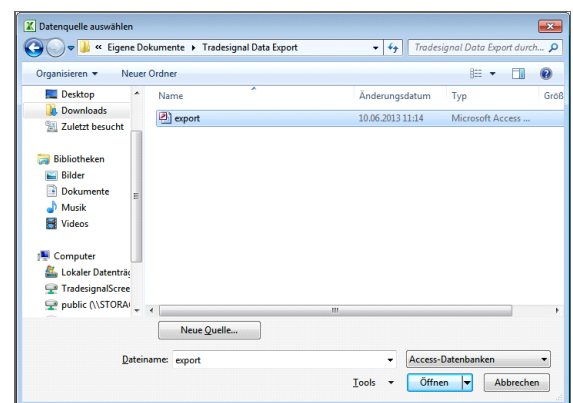
1. Make sure that COM-API extensions are enabled, *File > Options > Advanced > Equilla Extension*.
2. Apply indicator *LiveExport* from the *Tradesignal* package to an instrument or indicator on a chart.
3. Set path and table name prefix in the properties of the indicator. You can use macros for some commonly used folders on your machine, see source code of the indicator for details.



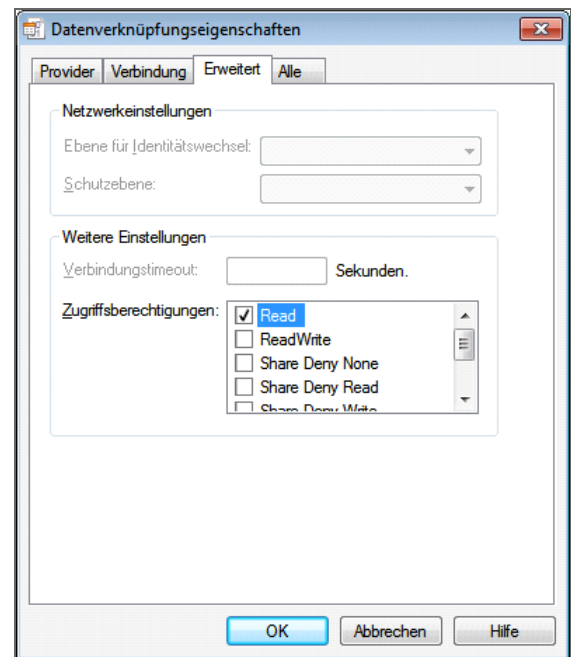
Equilla Extension options in Tradesignal

ACCESSING DATABASE FROM EXCEL 2010

1. *Data > From Access* on the ribbon allows you to pick the database file and table.
2. Make sure to pick *DB_MODE_READ* to open the database as read-only, otherwise Tradesignal will not be able to write to this database.
3. In the properties of the connection, you can enable periodic refresh and change the refresh interval.



Open Mode in Excel 2010



Connection Properties in Excel 2010

EXPORTING PRICE DATA

If your data provider allows exporting of price data, you can export data displayed in Tradesignal by

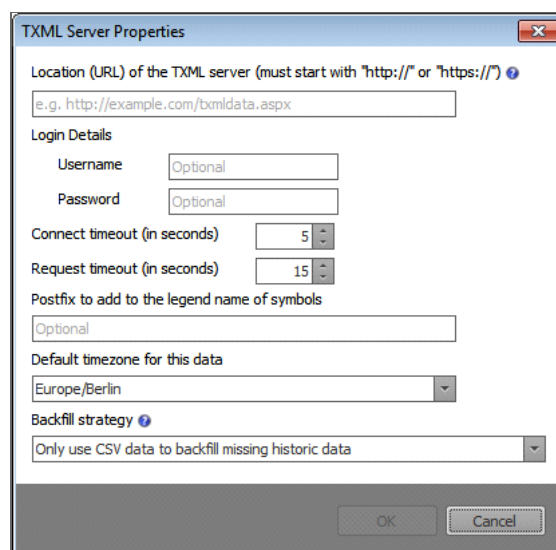
- dragging the symbol from the chart legend to another application which can handle CSV data input, e.g. most spreadsheet applications
- or by selecting the **Home > Data > Export Data** found in the Toolbar.
- The file format for export is CSV which can be parsed by a wide range of applications and is also human readable.

XML DATA SERVERS

You can integrate external data sources into Tradesignal. The following reasons work in their favor:

- Extension of histories: If you want to extend the history of your data provider with an external data source.
- Exchange historic data: If you want to substitute the histories of your data provider - partially or completely - by using better data from your external source.
- Extend the "data universe": If you want to use external data that your normal data provider doesn't offer.

Note: To connect the XML data server, a valid, special product license is necessary.



XML Data Server

PROCESSING PERSONAL DATA

For the professional use of external data sources, data banks are used that assemble and providing time histories. These data servers offer the data in a standardized XML format which can be interpreted and displayed by Tradesignal.

Usually, such a server is situated in the intranet, but since it is a normal web service, a use via internet is possible as well. Several data sources or data servers can be used.

In the **Options**, menu entry **Datafeeds**, you can configure up to ten external data sources. Instructions can be found at Additional Data Sources.

USING PROPRIETARY TIMESERIES IN TRADESIGNAL

In Tradesignal, trading houses or other companies that assemble data histories of their own are given the ability to display and chart these histories via the XML data server.

DATA INTERFACE

The data server interface offers an (internal) Search API, a Master Data API and a History API.

- Search API: Via this API, Tradesignal connects to the data source of the data server. The search is integrated so that the normal search functionality of the chart application are available.
- Master Data API: Via this API, master data like currencies, pieces, names, units etc. are available.
- History API: This API offers the complete data history of the requested instrument. The interface offers the same time periods as the data source. In case the requested history is not available on the server, the time history can also be accumulated in

Tradesignal. Due to a possible decrease in performance, however, the most important periods should always be pre-cumulated in the data source.

DATA UPDATES

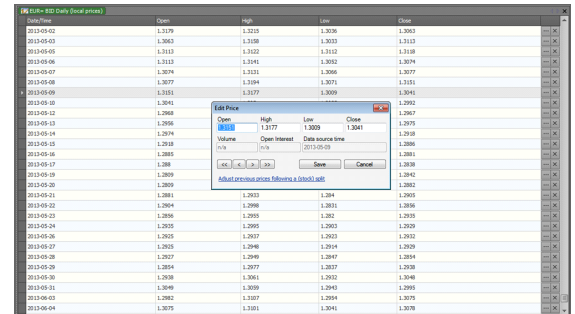
Since the XML data server does not "push" the data, Tradesignal has to access the data server at regular time intervals to pull current data. You can set these time intervals to the second in Tradesignal, even for each stock if desired. If the data have changed since the latest update, Tradesignal only pulls the data difference to optimize server and client performance and to minimize the necessary data volume.

PRICE EDITOR

Price data can be obtained from various sources. On the one hand, data is supplied to the software from the default data provider via the Internet, on the other hand data can be imported via a data interface, as text files or in other data formats.

When receiving data, errors may happen and make it necessary to edit the price data manually. Use the Price Editor to:

- enter stock splits retroactively
- rectify faulty ticks and corrupt data records



Price Editor

The Price Editor is available in the Chart, Scanner and Portfolio.

OPEN THE PRICE EDITOR

- Click on the button **Edit Prices** under *Home > Data* on the Toolbar.
- Alternatively, open the context menu of the documents and select **Edit Price**.

The Price Editor opens. The *Prices* group appears in the toolbar.

EDIT DATA

CHANGE DATA

You can edit the values of a line (single trading period).

- Double-click the line.
- Alternatively, right-click the line and select **Edit** from the context menu.

DELETE DATA

You can delete lines (single trading periods) from the price table.

- Select one or more lines and press the **Del** key.
- Alternatively, right-click the selected lines and select **Delete Row(s)** from the context menu.

Depending on file size, the process may take several minutes, as backups are created.

ENTER STOCK SPLITS

You can enter stock splits manually, if the data provider offers no corrected data or if you downloaded data without split correction from the Internet.

1. Select the lines you want to edit.
2. Open the split function by clicking the **Split** button in the symbol bar or selecting the entry **Split** in the context menu.

Select a Data Adjustment Mode:

- **Absolute** - The value is subtracted from the selected prices.
- **Relative** - Here you can enter a multiplier or the split ratio. Many stocks are split in a 1:2 or 1:3 ratio. The preview window displays the effects of the entered split settings on example data.

Click **OK** to save the settings and start the process. The data is changed. This may take a few minutes.

RESET CHANGES

Changed or deleted prices can be reset to their original values at any time (with the exception of DataConnect data). Tradesignal keeps backups of the data.

1. Select one or more lines with changed data. Changed data are underlined in orange in the table.
2. Open the context menu and select **Reset Edited Data**. The program restores the backup and overwrites the changes.

DIFFERENCE BETWEEN LOCAL DATA AND TRADESIGNAL DATACONNECT

Tradesignal DataConnect is a product from Tradesignal that collects and organizes data from many commercial data feeds. It is only relevant to Tradesignal institutional users.

The functions described above are available for local data or data offered from the provider, and also for the price data of DataConnect.

The sole difference is that in the case of DataConnect data, no backup and data restoration is available. If you want to change DataConnect data, save the data as personal backup before editing it.

⋮ MISCELLANEOUS

REUTERS DATA

SYMBOL SYNTAX

Reuters Symbol syntax differs from the Bloomberg syntax. In general, you only have to enter the symbol short code in the command line, for example

- short code + exchange (e.g. Siegn.DE)

The short code behind the colon refers to the exchange, e.g. the following:

- .DE - Xetra
- .FRA – Frankfurt
- .MU - Munich
- .STU – Stuttgart

In addition to those short codes, you can add more commands, e.g. period information.

For example: Siegn.de 5m (opens Siemens at xetra in 5minute period)

Continuous Future contracts are indicated with "c1", e.g. FDXc1 for dax future or FGBLc1 for Bund future.

Single contracts are entered as follows:

- short code + expiry month + expiry year

e.g. for dax future dec. 2007, it would be FDXZ7

Indices always have a dot before the short code, for example:

- .GDAXI for dax index
- .DJI for dow jones index
- .GSPC for s&p index

A list of all mayor symbol codes you can find here: **Reuters Symbolists**

SYMBOL LISTS

When starting TradeSignal for the very first time, you are asked if you want to receive lists from your data provider. If you confirm this, you will use the Symbol List wizard for the first time. However, you can start the wizard any time to create a new list.

- In the toolbox, click the Symbol Lists button.
- In the Related Task area, click on New Symbol List.

The Symbol List wizard opens. Choose **Predefined Symbol List from Your Data Provider**. On the upper left side you will find a search box. Enter the RIC of the wanted index and hit *Search*. Now the result will be displayed. Make a check into the checkbox next to the result and hit *OK*. Now the new Symbol List will be added to your list.

Use the RIC, you will use to open the Index as chart.

Please use:

.GDAXI for the Dax 30
.MDAXI for the MDax
.TECDAX for the TecDax
.FTSE for the FTSE 100 Index
.FCHI for the CAC 40 Index
.BFX for the Bell 20 Index
.AEX for the AEX Amsterdam Index
.IBEX for the Ibex 35 Index
.DJI for the Dow Jones Industrial Index
.IXIC for the Nasdaq Composite Index
.SPX for the S&P 500 Index
.BVSP for the Ibovespa - Bovespa Index
.HSI for the Hang Seng Index
.N225 for the Nikkei Stock Average 225
.SSMI for the SMI Index
.MIBTEL for the Italian Mibtel
.SMSI for the Madrid General Index
.OMXSPI for the Stockholm All Share Index
.OMXHPI for the Helsinki All Share Index
.OMXC20 for the Copenhagen 20 Index
.OSEAX for the Oslo Exchange All-share Index
.BUX for the Budapest Stock Exchange Index
.STOXX50 for the DJ Stoxx 50 ETF
.STOXX50E for the DJ Euro Stoxx 50 ETF
.AORD for the All Ordinaries Index
.KS11 for the KOSPI Index
.KLSE for the KLSE Composite Index
.SETI for the SET Composite Index
.JKSE for the Jakarta Composite
.PSI for the PSE Composite Index
.KSE for the Karachi SE 100 Index
.SSEC for the Shanghai Composite Index

BLOOMBERG DATA

SYMBOL SYNTAX

Bloomberg Symbol syntax differs from the Reuters syntax. In general, you only have to enter the symbol short code in the command line, for example

- short code + security type (e.g. SIE EQUITY)

Other short codes are:

- GOVT for government bonds
- CORP for corporate bonds
- CRNCY for currency
- INDEX for indices
- COMDTY for futures

Due to the fact that instruments are traded on different exchanges, you have to enter additional commands, otherwise the most profitable exchange will be offered automatically.

- short code + exchange + security type (e.g. SIE GY EQUITY for xetra)

In addition to those short codes, you can add more commands, e.g. period information.

For example:

- short code + exchange + security type + period (e.g. SIE GY EQUITY 5m opens Siemens on xetra in 5min. period)

Continuous Future contracts are indicated with "1", e.g. GX1 COMDTY for dax future or RX1 COMDTY for Bund future.

Single contracts are entered as follows:

- short code + expiry month + expiry year + security type

e.g. for dax future Dec. 2007, it would be GXZ7 COMDTY

You can recall more short codes and infos within the Bloomberg terminal. Enter the commands as follows:

- DOCS + Go formal documents
- BLP + Go Bloomberg Launch Pad
- PDF + Go personal default settings
- BBXL + Go all about Bloomberg und Excel G + Go

A list of all mayor symbol codes you can find here: **Bloomberg Symbolists**

DATAHISTORY

All data comes directly from Bloomberg. There will be no local caching or saving.

Generally there will be the following data history be available:

-	Daily	Intraday	Ticks
Stocks	20 - 25 Years	50 Days	5 Days
Futures (adj.)	ca. 15 Years	50 Days	50 Days
Indizes Dt	up to 1960 (Dax)	50 Days	50 Days
Indizes US	up to 1900 (DJI)	50 Days	50 Days
Indizes US	up to 1980 (NDX)	50 Days	50 Days

SYMBOL LISTS

When starting TradeSignal for the very first time, you are asked if you want to receive lists from your data provider. If you confirm this, you will use the Symbol List wizard for the first time. However, you can start the wizard any time to create a new list.

- In the toolbox, click the Symbol Lists button.
- In the Related Task area, click on New Symbol List.

The Symbol List wizard opens. Choose **Predefined Symbol List from Your Data Provider**. On the upper left side you will find a search box. Enter the shortcode of the wanted index and hit *Search*. Now the result will be displayed. Make a check into the checkbox next to the result and hit *OK*. Now the new Symbol List will be added to your list.

Use the shortcode, you will use to open the Index as chart.

Please use:

DAX for the Dax 30

MDAX for the M Dax

TDXP for the TecDax

UKX for the FTSE 100 Index

CAC for the CAC 40 Index

BEL20 for the Bell 20 Index

AEX for the AEX Amsterdam Index

IBEX for the Ibex 35 Index

INDU for the Dow Jones Industrial Index

CCMP for the Nasdaq Composite Index

SPX for the S&P 500 Index

SPTSX for the TSX COMPOSITE INDEX

IBOV for the Ibovespa - Bovespa Index

HSI for the Hang Seng Index

NKY for the Nikkei Stock Average 225

SMI for the SMI Index

ATX for the Österreichischen ATX Index

MIB30 for the Italian Mibtel
MADX for the Madrid General Index
OMX for the Stockholm All Share Index
HEX for the Helsinki All Share Index
KFX for the Copenhagen 20 Index
OBX for the Oslo Exchange All-share Index
BUX for the Budapest Stock Exchange Index
SX5P for the DJ Stoxx 50 ETF
SX5E for the DJ Euro Stoxx 50 ETF
AS30 for the Australian All Ordinaries Index
KOSPI for the South Korean KOSPI Index
KLCI for the KLSE Composite Index
SET for the Thai SET Composite Index
JCI for the Jakarta Composite
PCOMP for the Philippine PSE Composite Index
KSE100 for the Karachi SE 100 Index
SHCOMP for the Shanghai Composite Index

All material included in this document is for informational purposes only. The information published should not be considered as advice (financial or otherwise) and is not intended to replace consultation with a qualified professional.