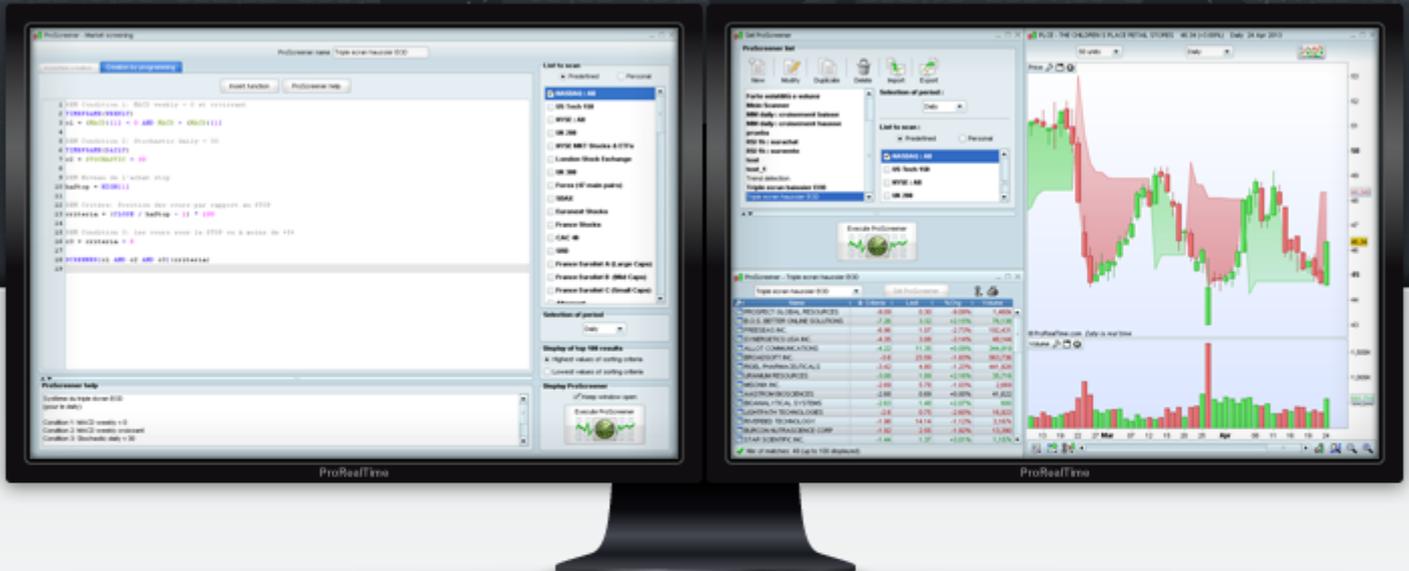




## Programming Guide Market Scans (ProScreener)



ProScreener is a powerful scanning tool that allows you to scan entire equity and forex markets for instruments that match your investment criteria.

Follow [ProRealTime Programming](#) on Google+ for updates about ProRealTime programming languages.

# TABLE OF CONTENTS

 <b>Introduction to ProScreener</b>	<b>1</b>
 <b>Chapter I: Fundamentals</b>	<b>2</b>
➔ Accessing ProScreener.....	2
➔ Using Top Movers (predefined scanning programs).....	6
 <b>Chapter II: Programming ProScreener</b>	<b>7</b>
➔ Searching and filtering results.....	7
➔ Volume Estimation.....	9
➔ Multi-period scanning.....	9
➔ Multi-security scan.....	10
 <b>Chapitre III: Practical Applications</b>	<b>12</b>
➔ Simple Examples.....	12
> RSI 1 hour: Oversold.....	12
> RSI 1 hour: Overbought.....	12
> Bullish moving average crossover.....	12
> Bearish moving average crossing.....	13
➔ More elaborate examples.....	13
> RSI and bullish reversal.....	13
> RSI and bearish reversal.....	14
> Bullish Engulfing with trend verification.....	14
> Bearish Engulfing with trend verification.....	15
> Triple bullish screen.....	16
> Triple bearish screen.....	17
 <b>Glossary</b>	<b>18</b>

*Warning: ProRealTime does not provide investment advisory services. This document is not in any case personal or financial advice nor a solicitation to buy or sell any financial instrument. The example codes shown in this manual are for learning purposes only. You are free to determine all criteria for your own trading. Past performance is not indicative of future results. Any trading system may expose you to a risk of loss greater than your initial investment.*

## Introduction to ProScreener

ProScreener is a powerful scanning tool that will allow you to scan equity markets and the forex for securities and currency pairs:

- meeting one or multiple customizable conditions defined by you
- in one timescale view or in multiple different timescale views (ex: 1 minute and 1 hour)
- in real-time or on the close of each bar with tick-by-tick precision

ProScreener uses ProBuilder language (you are advised to also consult the [ProBuilder manual](#)) with extensions that apply exclusively to filtering conditions.

The scans can be done in real-time or on the close of the current bar with the following timeframes:

- 1 minute
- 2 minutes
- 3 minutes
- 5 minutes
- 10 minutes
- 15 minutes
- 30 minutes
- 1 hour
- 2 hours
- 3 hours
- 4 hours
- Daily
- Weekly

The results of a ProScreener scan are displayed in a list of the top 50 securities meeting the conditions defined by the user on the basis of the sorting criteria, which updates in real-time.

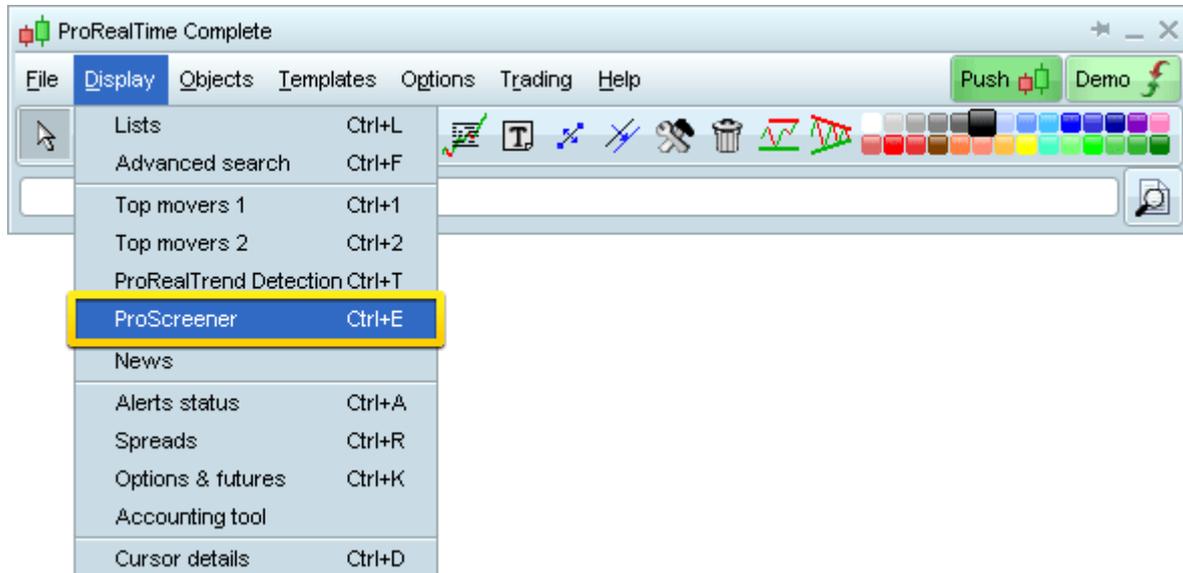
This guide is written as a continuation of the [ProBuilder Manual](#) but may also be consulted independently. The goal is to clearly explain how to best create ProScreeners, with all information related to commands with ProScreener and concrete examples. At the end, you will find a glossary with all the commands usable within ProScreener.

We hope you will enjoy the manual!

## Chapter 1: Fundamentals

### Accessing ProScreener

You can access ProScreener by clicking on “Display” and then “ProScreener as shown below:

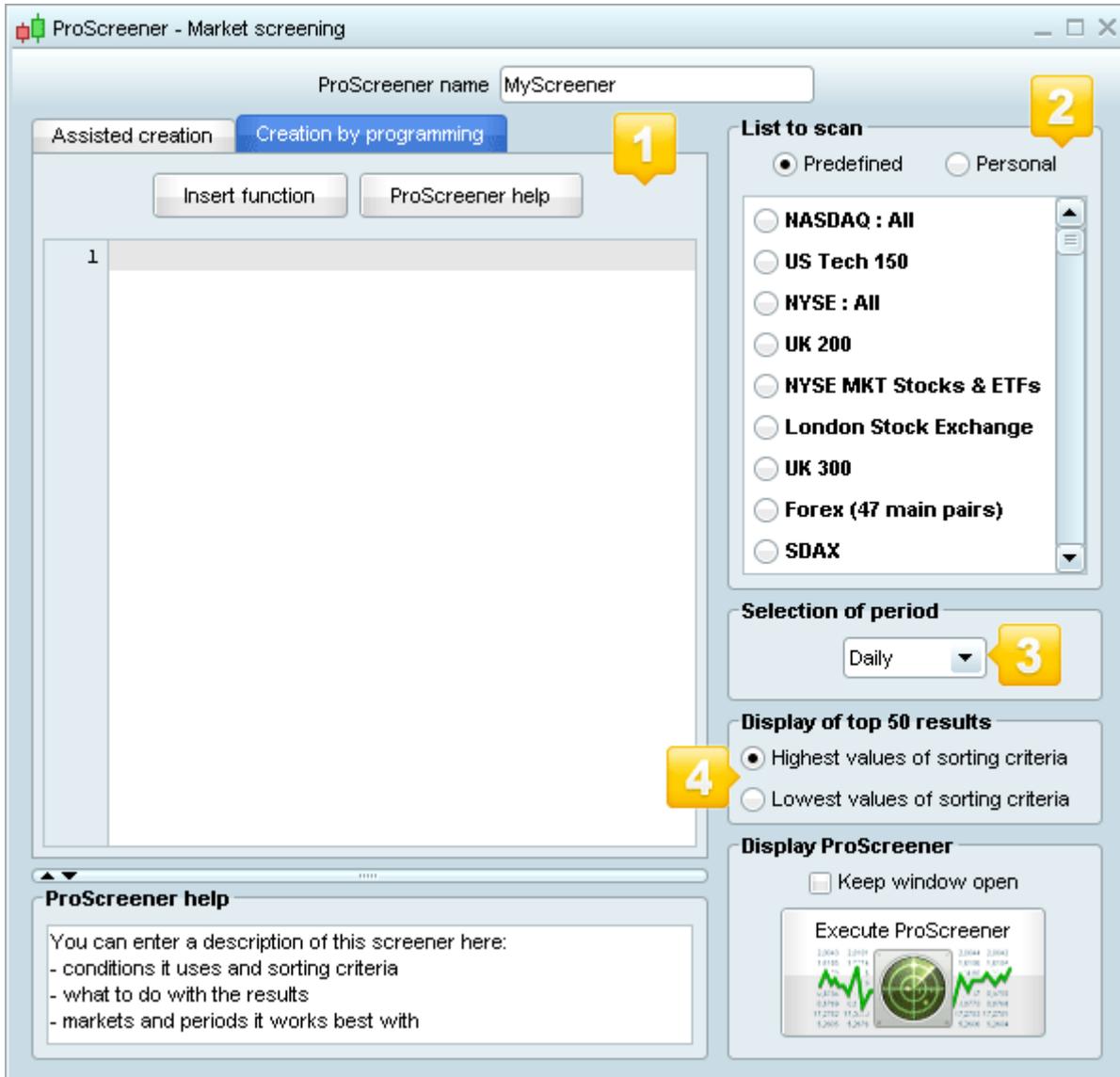


The ProScreener window will then appear, which allows you to select a ProScreener that is already created or to create a new ProScreener. To do this, click "Set ProScreener" and then "New ProScreener". From this window, you can choose 2 ways to create a new ProScreener:

- "Assisted Creation" mode which allows you to define the conditions of the ProScreener by simply point-and-clicking on one or more charts. To learn how to do this very easy assisted creation, please watch the video tutorial: "[ProScreener: Real-time scan with multiple conditions without writing a single line of code](#)".
- "Creation by Programming" that allows you to create more complex codes and define the parameters of your ProScreener.

The Creation by Programming window is made of 4 sections:

1. Programming Zone
2. Selection of List
3. Selection of Period
4. Sorting Method

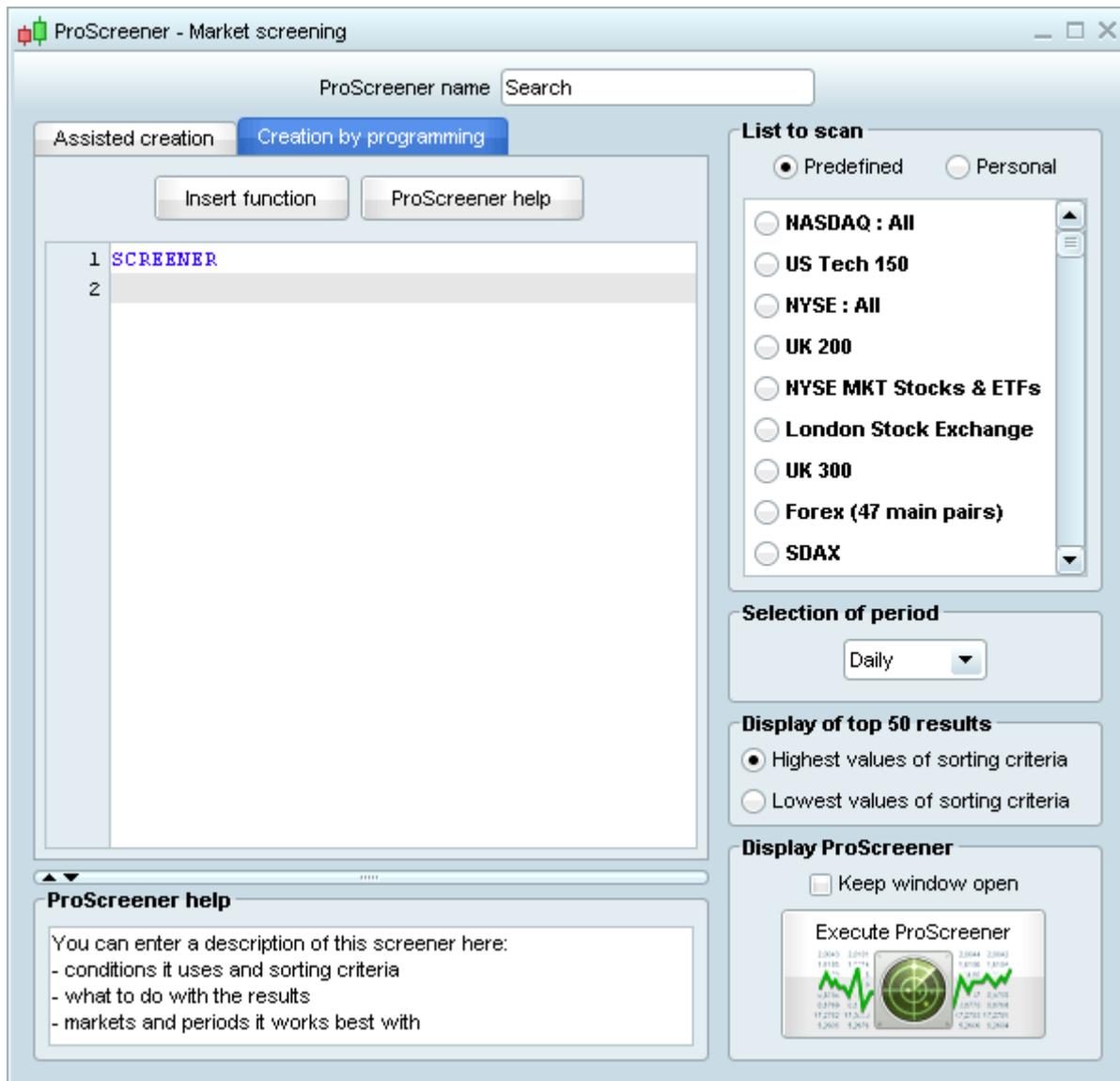


The first section (**Programming of ProScreener**), allows you to:

- Program a ProScreener directly in the text zone or
- Use "Insert function" to open the function library and insert a function from the library. The function library is divided into categories to help you quickly find the function you need and insert it with correct syntax. It also contains help text related to each function.

If you click "Insert Function" you will notice a special category of "ProScreener commands", which are for use only within ProScreener.

Choose the command "**SCREENER**" whose purpose is to execute the screener and click OK. This will insert the screener command in your program.



The "**SCREENER**" command defines what conditions to scan for. The last line of every ProScreener must be the "**SCREENER**" command.

Suppose we want to look for all of the NASDAQ stocks for which the open of the current bar is greater than the close of the previous bar. In our program, we will write the following in the programming zone:

```
c1 = (Open > Close[1])
SCREENER[c1]
```

When the code has been written we will go to the **second section (Selection of List)** allows you to select the list to scan on. We can choose to scan on a predefined list or a personal list.

*Note: The scan can only be done on securities within the same market. If you want to apply ProScreener to a personal list, you need to make sure that all the securities in the personal list are part of the same market (ex: all forex or all NASDAQ).*

The **third section (Selection of Period)** allows you to define the time period used in the ProScreener. The period used for the detection is important because the conditions you are looking for in one chart timeframe may be different in a different time frame.

The **fourth section (Display top 50 results)** concerns the sorting criteria of the results (See section: "[Searching and filtering results](#)"). With conditions that have very little restrictions, the results of a ProScreener may be very numerous (several thousand on the NASDAQ for example). In this case, you need to filter the results with a chosen criteria and choose to display:

- the results with the 50 highest values (of the sorting criteria)
- the results with the 50 lowest values (of the sorting criteria)

For Example: Imagine your ProScreener searches for the NASDAQ stocks with volume higher than 20 000 on the daily chart. The number of results will logically exceed 50. In your program, you can define volume as the sorting criteria and choose highest 50 values of sorting criteria (which will return the securities with TOP volume first) or lowest values of sorting criteria (which will return the securities with LOWEST volume still meeting the conditions of the screener first (those closest to 20 000)).

When you have set up the 4 sections as you want, click "Execute ProScreener" to validate the creation or modification of your ProScreener. If there are results, they will be displayed in the table as shown below.

The screenshot shows the ProScreener - Search window with a table of results. The table has columns for Name, Last, %Chg, Volume, and Prev. The results are sorted by Volume in descending order. A status bar at the bottom indicates 'Nbr of matches: 1,722 (up to 50 displayed)'.

Name	Last	%Chg	Volume	Prev
FBR & CO	20.39	+11.60%	92,487	18.27
PCM INC.	7.21	+6.03%	39,266	6.80
JOE S JEANS INC.	1.77	+5.99%	1,136k	1.67
PLURISTEM THERAPEUTICS	3.06	+4.44%	245,651	2.93
COGO GROUP INC.	1.90	+4.40%	33,471	1.82
RESEARCH IN MOTION	14.90	+3.98%	24,901k	14.33
E2OPEN INC.	12.96	+3.85%	208,324	12.48
CITI TRENDS INC.	11.27	+3.21%	69,814	10.92
LRAD CORP.	0.96	+2.13%	65,226	0.94
FIRST INTERSTATE BANCSYSTEM	19.21	+1.69%	209,159	18.89
RUTH S HOSPITALITY GROUP	9.85	+1.65%	271,180	9.69
WESTMORELAND COAL CO.	11.57	+1.58%	41,407	11.39
STARZ SERIES A LIBERTY CAPITAL	23.10	+1.32%	1,278k	22.80
AVEO PHARMACEUTICALS	8.27	+1.22%	2,512k	8.17
SEARS HOLDINGS CORP.	48.34	+1.19%	609,131	47.77
CHINA HOUSING & LAND DEVELOPMENT	1.78	+1.14%	67,161	1.76
CRIMSON EXPLORATION	2.86	+1.06%	54,297	2.83
INFORMATION SERVICES GROUP	1.91	+1.06%	24,672	1.89
BLACK DIAMOND INC.	9.85	+0.92%	135,311	9.76
MEASUREMENT SPECIALTIES	42.02	+0.86%	46,181	41.66
COFFEE HOLDING CO.	6.54	+0.77%	61,269	6.49
CHINA LODGING GROUP	15.32	+0.66%	150,482	15.22
STUDENT TRANSPORTATION	6.41	+0.63%	124,267	6.37
PRIMORIS SERVICES	21.76	+0.60%	363,042	21.63
PENNANTPARK INVESTMENT	11.28	+0.53%	447,005	11.22

✓ Nbr of matches: 1,722 (up to 50 displayed)

## Using Top Movers (predefined scanning programs)

ProRealTime includes 2 Top Movers windows which may be viewed simultaneously with a ProScreener window. The top movers windows include several types of predefined scans which are easy to view and understand. The two top movers windows may be viewed simultaneously to allow you to search at the same time for top movers in two directions (ex: Top movers up and top movers down) in the case of the variations tab.

The available tabs of the Top Movers tool include and allow you to detect:

- Variations, gaps and abnormal volume
- Pre-opening variations
- Candlestick form detection
- High order book spread (difference between bid and ask) or high increase in bid or ask volume (up or down)

Top Movers - Variations - NASDAQ: All - Up

Exchange: NASDAQ: All | Criteria: Up | Since: Yesterday | Nbr of matches: 29

Name	%Chg Yest.	Volume	Last	Price ref
FX ENERGY INC.	+27.04%	1,697k	3.43	2.7
GENERAL FINANCE	+25.00%	10,000	0.15	0.12
INSMED INC.	+17.64%	598,887	8.87	7.54
REAL GOODS SOLAR	+16.77%	1,074k	1.88	1.61
SUCAMPO PHARMACEUTICALS	+16.67%	1,524k	9.87	8.46
CIMATRON LTD.	+16.42%	1,252k	7.80	6.7
FEDERAL-MOGUL CORP.	+15.63%	448,401	5.77	4.99
ZIONS BANCORP.	+14.16%	2,500	2.50	2.19
IROBOT CORP.	+13.71%	2,299k	28.04	24.66
ECOTALITY INC.	+13.42%	791,659	1.69	1.49
VELTI PLC	+13.04%	2,566k	1.82	1.61
POINT.360	+12.39%	41,911	1.27	1.13
CHINA INFORMATION TECHNOLOGY	+12.17%	329,641	1.29	1.15
FIRST SOLAR INC.	+11.82%	17,154k	45.11	40.34
EDUCATION MANAGEMENT	+11.81%	173,684	5.68	5.08
BIOCRYSST PHARMACEUTICALS	+11.67%	4,823k	2.01	1.8
FBR & CO	+11.60%	92,487	20.39	18.27
BROADWAY FINANCIAL	+11.25%	200	0.89	0.8
SCG FINANCIAL ACQUISITION CORP	+11.13%	12,856	12.78	11.5
GUANWEI RECYCLING	+10.74%	33,315	1.34	1.21
SUNPOWER CORP.	+10.45%	7,366k	12.26	11.1
RUBICON TECHNOLOGY	+10.31%	1,001k	7.06	6.4
SOLAZYME INC.	+10.11%	1,024k	8.71	7.91
OSI SYSTEMS INC.	+9.76%	1,654k	58.70	53.48
COMSTOCK HOLDING COMPANIES	+9.49%	647,903	3.00	2.74

To learn more about the Top Movers predefined scanning tools, please watch the related videos on our video tutorials page including "[Top movers scanner: detect price and volume top movers or opening gaps](#)"

## Chapter II: Programming ProScreener

In this chapter we will illustrate the 5 commands in the ProBuilder language that are only for use in the ProScreener module and allow you to do scans. These commands are displayed in the "ProScreener" commands section when you press the "Insert Function" button in the programming zone. We will look at:

- Searching and filtering results
- Volume estimation
- Multi-period scanning
- Multi-instrument scanning

### Searching and filtering results

---

The "**SCREENER**" command defines what conditions to scan for. The last line of every ProScreener must begin with "**SCREENER**". Like the return function in ProBuilder, it can be followed by other commands. The syntax for "**SCREENER**" is as follows:

#### **SCREENER[Condition]**

Example:

```
c1 = (Close < BollingerDown[10](Close))
SCREENER[c1]
```

Let's find all of the instruments in a market for which the closing price is strictly less than the lower Bollinger Band. Bollinger bands are calculated using 10 periods and applied to the closing price.

It is possible to scan for more than one condition with AND (both conditions must be met) or OR (at least one condition must be met). To do this in ProScreener, the syntax is:

#### **SCREENER[Condition1 AND Condition2]**

OR

#### **SCREENER[Condition1 OR Condition2]**

Example:

Lets search for securities for which price is greater than the upper Bollinger band and have an increasing trend shown by the fact that the close is greater than the open and the 7-period moving average is above the 23-period moving average.

```
REM Close is higher than the Upper Bollinger band of 20 periods calculated on the close
Condition1 = (Close > BollingerUp[20](Close))
REM Close > Open
Condition2 = Close > Open
REM MA7 calculated on close > MA23 calculated on close
Condition3 = (Average[7](Close) > Average[23](Close))
SCREENER[Condition1 AND Condition2 AND Condition3]
```

It is also possible to define a constant or an indicator that will also work as a filter and sorting criteria for the results:

- If the scan returns more than 50 results, the indicator filters the securities to display. In this case, the 4th section discussed above "Display top 50 results" allows you to define whether you want to display the results with the highest or lowest values of the sorting criteria (see [description above](#)).
- If the scan returns less than 50 results, the scanner will sort the results based on the sorting criteria that you have defined.

The syntax to use a constant as a sorting criteria is:

**SCREENER[c1](TypicalPrice)**

To use a predefined indicator as a sorting criteria, it must be first stocked in a variable (here we will call it "Criteria"):

**Criteria = RSI[14](Close)**  
**SCREENER[c1](Criteria)**

For example, we might have defined criteria to be the moving average of 20 periods on the close by entering  
 Criteria = Average[20](Close)

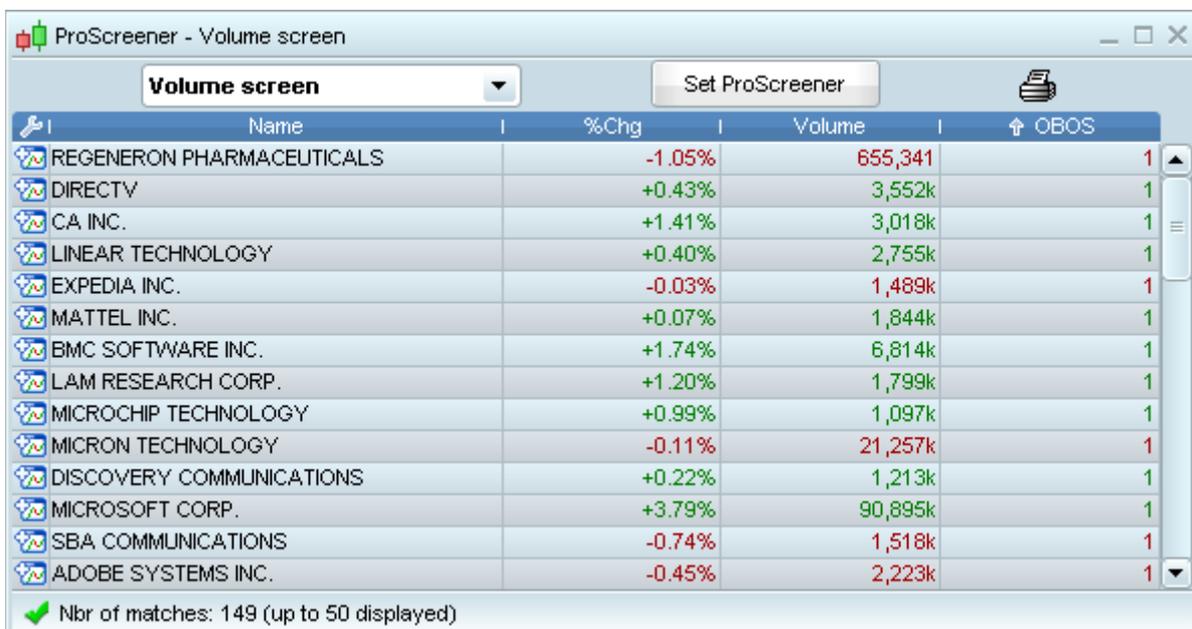
If we call a personalized indicator (indicator written in ProBuilder language), we need to use the "CALL", command as defined in the [ProBuilder Manual](#).

**MyRSI = CALL "RSI" [14]**  
**SCREENER[c1](MyRSI)**

Example:

Let's find securities whose volume is more than 50000. Then lets sort the results by whether the RSI is overbought or oversold. To measure whether the RSI is overbought or oversold, we create a binary indicator which is equal to 1 if its greater than 70 and -1 if its less than 30. We will display the result of the binary indicator in the sorting criteria column.

```
c1 = Volume > 50000
IF RSI[20](Close) > 70 THEN
    Criteria = 1
ELSIF RSI[20](Close) < 30 THEN
    Criteria = -1
ENDIF
SCREENER[c1](Criteria AS "OBOS")
```



It is possible to not use the parenthesis if we use the "SORT BY" command instead:

**SCREENER[c1] SORT BY TypicalPrice**

If we want to give a personalized name to this column, we should use the "AS" function as illustrated here:

**SCREENER[c1](TypicalPrice AS "typical price")**

## Volume Estimation

The "**EstimatedVolume**" command allows us to do a linear estimation of what the ending volume of the current bar will be. Estimated Volume is calculated in the following way:

$$\text{EstimatedVolume} = \text{Volume} * \text{Multiplier}$$

where

$$\text{Multiplier} = \text{Unit of time} / \text{time that the current candlestick has been active}$$

This command is particularly interesting to compare estimated volume to actual volume.

Lets look at an example:

In 10-minute view, if 1 minute has passed, we can estimate that the volume for the candle at the end will be 10 times the current volume for the current candle.

For example, let's calculate the ratio of today's estimated volume to the previous day's volume:

```
REM Vol0 estimates the volume for the current bar
Vol0 = EstimatedVolume
REM Vol1 = the volume of the previous bar
Vol1 = Volume[1]
REM The screener will organize the results by the ratio of estimated volume of the
current bar to actual volume of the previous bar.
SCREENER (Vol0 / Vol1 AS "VRatio")
```

## Multi-period scanning

Its possible to do a ProScreener with multiple conditions and on multiple time periods. This enables you to check for example if your a condition is true on several different charting timeframes which you may want to look at (ex: short and long-term chart). The command to use is "**TIMEFRAME**" and its syntax is as follows:

### **TIMEFRAME**(unit of time of your choice)

The timeframes available in ProScreener are listed below:

TIMEFRAME	CODE
1 minute	1 minute
2 minutes	2 minutes
3 minutes	3 minutes
5 minutes	5 minutes
10 minutes	10 minutes
15 minutes	15 minutes
30 minutes	30 minutes
1 hour	1 hour
2 hours	2 hours
3 hours	3 hours
4 hours	4 hours
Daily	daily
Weekly	weekly

The conditions following the "TIMEFRAME" command will be scanned for only in the period indicated. It is possible to use multiple "TIMEFRAME" lines in the same program to do a multi-period scan.

Example:

We want to find all the NASDAQ stocks corresponding to the criteria below:

- In the weekly timeframe, the Williams %R applied to close of 14-periods is between 0 and -20
- In the 30-minute timeframe, the EMA of 20 periods applied to close recently crossed over the EMA of 12 periods.

This example has the goal of detecting overbought values on the weekly timeframe with a bearish trend on the 30-minute timeframe. Here is the code for this example ProScreener:

```
TIMEFRAME(weekly)
Condition1 = Williams[14](Close) < 0 AND Williams[14](Close) > -20
TIMEFRAME(30 minutes)
Condition2 = ExponentialAverage[20](Close) CROSSES OVER ExponentialAverage[12](Close)
SCREENER[Condition1 AND Condition2]
```

## Multi-security scan

---

With ProScreener, we have seen that we need to specify a market or list on which to do our scan.

"EQUITYFRAME" allows us to scan for a criteria related to a specific security in the market we are scanning.

The syntax for "EQUITYFRAME" is:

**EQUITYFRAME("market name", "ticker")**

The command allows us to compare the results of the scan to a particular security, or construct a new variable for use later in the code, or also to construct a new indicator as a sorting criteria. This command has a few particularities:

- You must have real-time access for the market you are looking at
- Cannot be used with indices
- The security "ticker" referenced must be part of the market you are currently scanning

For example, if you want to display on 50 random values of the NASDAQ, the current price of MICROSOFT (ticker: MSFT) as the sorting criteria in the ProScreener table, you could use this code:

```
EQUITYFRAME("NASDAQ", "MSFT")
MyClose = Close
SCREENER(MyClose)
```

Les différents marchés correspondent à la syntaxe reportée dans le tableau ci-dessous:

CODE EQUITYFRAME	NOM MARCHÉ	CODE EQUITYFRAME	NOM MARCHÉ
Actions & ETFs AMEX	Actions "Amex"	Actions Italie	Actions "Italie"
Actions Australie	Actions "Australie"	LSE	London Stock Exchange
Actions Espagne	Actions "Espagne"	Actions NASDAQ	Actions "Nasdaq"
Actions Amsterdam	Euronext Amsterdam	Actions NYSE	Actions "Nyse"
Actions Belgique	Euronext Bruxelles	Actions Suisse VirtX	Actions "Suisse"
Actions Portugal	Euronext Lisbonne	Actions Suisse SWX	Actions "Suisse"
Actions France	Euronext Paris	Actions Allemagne	Actions "Allemagne"
Forex 47 paires	Forex		

Comme pour la commande "TIMEFRAME", tout code qui suivra "EQUITYFRAME" sera appliqué à la valeur spécifié par ce dernier. Pour revenir aux données du marché sélectionné dans l'interface ProScreener, nous pouvons écrire:

### EQUITYFRAME(default)

The 2 functions displayed above compliment eachother as the example below shows:

If you want to display in the ProScreener results table, the indicator that shows the difference between the close of the current bar for the currency pair EUR/USD (ticker "EURUSD") and the closing price of the results of the screener, you can do so in this way:

```
REM We begin by choosing the equityframe EURUSD and creating a variable to store its
closing price
EQUITYFRAME("Forex 47 Major Pairs","EURUSD")
MyClose = Close
REM We come back to the default equityframe for the market selected in the section
"Selection of List" (Forex in this case)
EQUITYFRAME(default)
REM We do the difference between the close of EURUSD and the close of the scan result
CloseVal = MyClose - Close
REM We filter the results by the difference between EURUSD and the price of the scan
result
SCREENER(CloseVal AS "MyIndicator")
```

Example: This screener allows us to visually display the correlation between a security and other securities in the same market. We calculate the ratio of the closing prices of 2 securities selected using "EQUITYFRAME". We then calculate the difference between the current level of the ratio and the ratio for the previous day.

```
TIMEFRAME(daily)
CloseVal = Close
EQUITYFRAME("NASDAQ","AMZN")
CloseInd = Close
EQUITYFRAME(default)
Ratio = (CloseVal / CloseInd) * 100
RelativeStrength = (Ratio - Ratio[1]) * 100
SCREENER(RelativeStrength AS "RelativeStrength")
```

## Chapitre III: Practical Applications

### Simple Examples

---

#### RSI 1 hour: Oversold

RSI is an overbought/oversold indicator that can predict trend reversals. In this example, we will scan for oversold securities. "Oversold" is defined by **RSI** < 30. This becomes more significant as the RSI approaches the 0 line.

We will construct a ProScreener that shows the securities where **RSI** < 30. This could be applied to an hourly timeframe.

##### Example: Oversold RSI

```
REM Calculate the 14-period RSI
MyRSI = RSI[14]
REM Filter: RSI < 30
Filter = MyRSI < 30
SCREENER[Filter](MyRSI AS "RSI")
```

#### RSI 1 hour: Overbought

Let's look for securities where **RSI** > 70 (overbought). The overbought signal becomes stronger as the RSI approaches 100.

This screener searches for securities with **RSI** > 70. It could be applied to an hourly timeframe.

##### Example: Overbought RSI

```
REM Calculate the 14-period RSI
MyRSI = RSI[14]
REM Filter: RSI > 70
Filter = MyRSI > 70
SCREENER[Filter](MyRSI AS "RSI")
```

#### Bullish moving average crossover

The bullish moving average crossover system is one of the most famous. It's based on the observation that a short-term moving average crosses over a long-term moving average when a bullish trend appears.

We are going to build a ProScreener that detects securities with a 20-period MA crossing over a 50-period MA. Furthermore, we calculate "momentum" which we will define as the difference between the 2 MAs in order to learn the power of the cross. If the number is close to 0, the cross is slower and less significant. The higher this number is, the stronger the crossing. A typical of a strong crossing is a relatively flat long-term MA and a very upward sloping short-term MA.

We will use this momentum as the sorting criteria of the screener, showing the highest values of TrendForce first.

##### Example: Detection of bullish crossover with the 20 and 50-period simple moving average

```
REM Calculate the short-term ma
MaShort = Average[20]
REM Calculate the long-term ma
MaLong = Average[50]
REM Determine the relative speed of the short MA to the long MA
Speed = Momentum(MaLong - MaShort) * 100 / Close
REM Detect the securities on which the crossover has just occurred, sort the results by Speed
Filter = MaShort CROSSES OVER MaLong
SCREENER[Filter](Speed AS "Dynamique")
```

## Bearish moving average crossing

This ProScreener shows us the securities for which the 20-period moving average crosses under the 50-period moving average.

We also use "momentum" in this example in the same way as in the previous example.

Example: Detection of bearish crossover with the 20 and 50-period simple moving average

```
REM Calculate the short-term ma
MaShort = Average[20]
REM Calculate the long-term ma
MaLong = Average[50]
REM Determine the relative speed of the short MA to the long MA
Speed = Momentum(MaShort - MaLong) * 100 / Close
REM Select securities when the cross occurs
Filter = MaShort CROSSES UNDER MaLong
SCREENER[Filter] (Speed AS "Dynamique")
```

## More elaborate examples

---

### RSI and bullish reversal

This system allows you to detect in real-time securities susceptible to make a bullish reversal (go from down trend to up-trend).

Traditionally analysts looked at simple indicators without considering all of the characteristics of the situation (because they did not have all the tools enabling them to do better).

With ProScreener, there are no more technical limitations. We can create a real system. To begin, we look for an oversold RSI that is increasing.

We create a filter on these 2 conditions which is written with one line of ProBuilder code:

```
Filter = RSI < 30 AND Momentum[1] (RSI) > 0
```

Now, we can look for the most interesting securities. These are the ones whose fall was most severe compared to their normal volatility.

Example: RSI and bullish reversal

```
REM Filter the securities whose RSI is oversold and in a reversal measured by the
momentum of the RSI.
// Filter: RSI < 30 and increasing
Filter = RSI < 30 AND Momentum[1] (RSI) > 0
REM Determine the force of the bearish trend
// Find the highest high of the last 20 bars.
Highest20 = highest[20] (High)
// Determine the decline since this period
Decline = Highest20 - Close
// Determine the normal volatility of the security (median of true range over the last 3
bars)
NormalV = summation[3] (TR) - highest[3] (TR) - lowest[3] (TR)
REM Display results. Sorting Criteria: Decline/NormalV (preceeding down trend force)
SCREENER[Filter] (Decline / NormalV AS "Down Trend Force")
```

## RSI and bearish reversal

This screener searches for an overbought RSI that is making a bearish reversal. As before, we create a filter:

```
Filter = RSI > 70 AND Momentum[1](RSI) < 0
```

In the same way as the previous ProScreener, we search for the securities for which the increase in price was the most marked compared to the normal volatility of the security.

### Example: RSI and bearish reversal

REM Filter the securities whose RSI is oversold and in a reversal, measured by the momentum of the RSI.

```
// Filter: RSI > 70 and decreasing
```

```
Filter = RSI > 70 AND Momentum[1](RSI) < 0
```

REM Determine the "force" of the bullish trend

```
// Find the lowest low of the last 20 bars
```

```
Lowest20 = lowest[20](Low)
```

```
// Determine the variation of price between the current price and the lowest low
```

```
Increase = Close - Lowest20
```

```
// Determine the normal volatility of the security (median of true range over the last 3 bars)
```

```
NormalV = summation[3](TR) - highest[3](TR) - lowest[3](TR)
```

REM Display results. Sorting Criteria: Increase/NormalV (preceeding up trend force)

```
SCREENER[Filter](Increase / NormalV AS "Up Trend Force")
```

## Bullish Engulfing with trend verification

It's easy to use ProScreener to detect many candlestick patterns. The basic top movers tool also lets you detect candlestick patterns, but if you look for them with ProScreener, you can have more control over the definition of the candlestick form and also add additional conditions as we will see in this example.

Now lets look at a ProScreener for the candlestick form "Bullish Engulfing". The bullish engulfing is one of the more powerful candlestick trend reversal patterns, but of course you must also look at the overall context (ex: existence of a prior down trend or not).

We can define a Bullish Engulfing in this way:

- Previous candlestick where Close < Open
- Open of current candle < Close of previous candle
- Current candle Close > Open of previous candle

These 3 conditions can be expressed in one line of code:

```
Filter = Close[1] < Open[1] AND Open < Close[1] AND Close > Open[1]
```

For a valid bullish engulfing to be detected, we want to also verify that a bearish trend existed prior to the appearance of the bullish engulfing. One way of doing this would be to use part of the previous code and tighten the time horizon. With the RSI 14 as examined in the previous bar, it was ok to look at the fall over the last 20 bars.

When looking for a reversal in the form of Japanese candlesticks, experience has shown that 8 bars is sufficient (these structures are more responsive and theoretically limited to 7 candles).

**Example: Bullish Engulfing with trend verification**

```

REM Determine the "force" of the preceding down trend.
// Find the highest high over the last 8bars
High8 = highest[8](High)
// Determine the decline since this point
Decline = High8 - Close
// Determine the normal volatility of the security (median of true range over the last 3
bars)
NormalV = summation[3](TR) - highest[3](TR) - lowest[3](TR)
REM Condition: Bullish Engulfing
Filter = Close[1] < Open[1] AND Open < Close[1] AND Close > Open[1]
REM Sorting criteria: Decline/NormalV (preceding down trend force)
SCREENER[Filter](Decline / NormalV AS "Down Trend Force")

```

**Bearish Engulfing with trend verification**

Let's look at another example with a Bearish Engulfing with trend verification.

A bearish engulfing can be defined as follows:

- Previous candle where Close > Open
- Open of current candle > Close of previous candle
- Close of current candle < Open of previous candle

These 3 conditions can be expressed in one line of code:

```
Filter = Close[1] > Open[1] AND Open > Close[1] AND Close < Open[1]
```

To detect the existence of a previous increasing trend, we use a code similar to the one above.

**Example: Bearish engulfing with trend verification**

```

REM Determine the "force" of the preceding up trend.
// Find the lowest point over the last 8 bars
Low8 = lowest[8](Low)
// Determine the increase since this point
Increase = Close - Low8
// Determine the normal volatility of the security (median of true range over the last 3
bars)
NormalV = summation[3](TR) - highest[3](TR) - lowest[3](TR)
REM Condition: Bearish engulfing Engulfing
Filter = Close[1] > Open[1] AND Open > Close[1] AND Close < Open[1]
REM Display results. Sorting Criteria: Increase/NormalV (preceding up trend force)
SCREENER[Filter](Increase / NormalV AS "Up Trend Force")

```

### Triple bullish screen

This example ProScreener is composed of 3 conditions in several units of time:

- **Condition 1:** MACD weekly < 0 and increasing.
- **Condition 2:** Stochastic Daily < 30.
- **Condition 3:** Price is less than previous day's high or no more than 5% above it.

The results will respect the conditions in all of the units of time specified (weekly and daily).

#### Example: Triple bullish screen

```
REM Condition 1 and 2: MACD weekly < 0 and increasing
TIMEFRAME(weekly)
MyMACD = MACD[12,26,9](Close)
c1 = MyMACD < 0 AND MyMACD > MyMACD[1]
REM Condition 2: Daily Stochastic < 30
TIMEFRAME(daily)
MySTO = Stochastic[14,3](Close)
c2 = MySTO < 30
REM Set Stop Level
MyStop = High[1]
REM Criteria: Proximity to the high of the previous day
Criteria = (Close / MyStop - 1) * 100
REM Condition 3: Price is less than previous day's high or no more than 5% above it.
c3 = Criteria < 5
SCREENER[c1 AND c2 AND c3](Criteria)
```

### Triple bearish screen

This ProScreener is made of 3 conditions using several units of time:

- **Condition 1:** MACD Weekly > 0 and decreasing
- **Condition 2:** Stochastic Daily > 70
- **Condition 3:** Price is greater than previous day's low or no more than 5% below it

We look for securities whose prices are the closest to the sell stop level recommended by the system. This level is the lowest point of the previous day.

The ProScreener displays the securities which are below this level and are still within -5% of it.

The results displayed respect the specified conditions in weekly and daily views.

### Example: Triple bearish screen

```
REM Condition 1: MACD weekly > 0 and decreasing
TIMEFRAME(weekly)
MyMACD = MACD[12,26,9](Close)
c1 = MyMACD > 0 AND MyMACD < MyMACD[1]
REM Condition 2: Stochastic daily > 70
TIMEFRAME(daily)
MySTO = Stochastic[14,3](Close)
c2 = MySTO > 70
REM Set Stop Level
MyStop = Low[1]
REM Sorting Criteria: Position of price with regard to stop level
Criteria = (Close / MyStop - 1) * 100
REM Condition 3: Price is greater than the stop level or less than 5% below it
c3 = Criteria > -5
SCREENER[c1 AND c2 AND c3](Criteria)
```

## Glossary

### A

CODE	SYNTAX	FUNCTION
ABS	ABS(a)	Mathematical function "Absolute Value" of a
AccumDistr	AccumDistr(close)	Classical Accumulation/Distribution indicator
ADX	ADX[N]	Indicator Average Directional Index or "ADX" of n periods
ADXR	ADXR[N]	Indicator Average Directional Index Rate or "ADXR" of n periods
AND	a AND b	Logical AND Operator
AroonDown	AroonDown[N]	Aroon Down indicator of n periods
AroonUp	AroonUp[N]	Aroon Up indicator of n periods
ATAN	ATAN(a)	Mathematical function "Arctangent" of a
AS	RETURN result AS "ResultName"	Instruction used to name a line or indicator displayed on chart. Used with "RETURN"
Average	Average[N](price)	Simple Moving Average of n periods
AverageTrueRange	AverageTrueRange[N](price)	"Average True Range" - True Range smoothed with the Wilder method

### B

CODE	SYNTAX	FUNCTION
BarIndex	BarIndex	Number of bars since the beginning of data loaded (in a chart in the case of a ProBuilder indicator or for a trading system in the case of ProBacktestProBacktest or ProInvest)
BollingerBandWidth	BollingerBandWidth[N](price)	Bollinger Bandwidth indicator
BollingerDown	BollingerDown[N](price)	Lower Bollinger band
BollingerUp	BollingerUp[N](price)	Upper Bollinger band
BREAK	(FOR...DO...BREAK...NEXT) or (WHILE...DO...BREAK...WEND)	Instruction forcing the exit of FOR loop or WHILE loop

**C**

CODE	SYNTAX	FUNCTION
CALL	myResult = CALL myFunction	Calls a user indicator to be used in the program you are coding
CCI	CCI[N](price) or CCI[N]	Commodity Channel Index indicator
ChaikinOsc	ChaikinOsc[Ch1, Ch2](price)	Chaikin oscillator
Chandle	Chandle[N](price)	Chande Momentum Oscillator
ChandeKrollStopUp	ChandeKrollStopUp[Pp, Qq, X]	Chande and Kroll Protection Stop on long positions
ChandeKrollStopDown	ChandeKrollStopDown[Pp, Qq, X]	Chande and Kroll Protection Stop on short positions
Close	Close[N]	Closing price of the current bar or of the n-th last bar
COLOURED	RETURN Result COLOURED(R,G,B)	Colors a curve with the color you defined using the RGB convention
COS	COS(a)	Cosine Function
CROSSES OVER	a CROSSES OVER b	Boolean Operator checking whether a curve has crossed over another one
CROSSES UNDER	a CROSSES UNDER b	Boolean Operator checking whether a curve has crossed under another one
cumsum	cumsum(price)	Sums a certain price on the whole data loaded
CurrentDayOfWeek	CurrentDayOfWeek	Represents the current day of the week
CurrentHour	CurrentHour	Represents the current hour
CurrentMinute	CurrentMinute	Represents the current minute
CurrentMonth	CurrentMonth	Represents the current month
CurrentSecond	CurrentSecond	Represents the current second
CurrentTime	CurrentTime	Represents the current time (HHMMSS)
CurrentYear	CurrentYear	Represents the current year
CustomClose	CustomClose[N]	Constant which is customizable in the settings window of the chart (default: Close)
Cycle	Cycle(price)	Cycle Indicator

**D**

CODE	SYNTAX	FUNCTION
Date	Date[N]	Reports the date of each bar loaded on the chart
Day	Day[N]	Reports the day of each bar loaded in the chart
Days	Days[N]	Counter of days since 1900
DayOfWeek	DayOfWeek[N]	Day of the week of each bar
DClose	DClose(N)	Close of the n-th day before the current one
DEMA	DEMA[N](price)	Double Exponential Moving Average
DHigh	DHigh(N)	High of the n-th bar before the current bar
DI	DI[N](price)	Represents DI+ minus DI-
DIminus	DIminus[N](price)	Represents the DI- indicator
DIplus	DIplus[N](price)	Represents the DI+ indicator
DLow	DLow(N)	Low of the n-th day before the current one
DO	See FOR and WHILE	Optional instruction in FOR loop and WHILE loop to define the loop action
DOpen	DOpen(N)	Open of the n-th day before the current one
DOWNTO	See FOR	Instruction used in FOR loop to process the loop with a descending order
DPO	DPO[N](price)	Detrended Price Oscillator

**E**

CODE	SYNTAX	FUNCTION
EaseOfMovement	EaseOfMovement[I]	Ease of Movement indicator
ELSE	See IF/THEN/ELSE/ENDIF	Instruction used to call the second condition of If-conditional statements
ELSEIF	See IF/THEN/ELSIF/ELSE/ENDIF	Stands for Else If (to be used inside of conditional loop)
EMV	EMV[N]	Ease of Movement Value indicator
EQUITYFRAME	EQUITYFRAME("market","ticker")	Condition related to a specific security in a specific market (ProScreener command only)
ENDIF	See IF/THEN/ELSE/ENDIF	Ending Instruction of IF-conditional statement
EndPointAverage	EndPointAverage[N](price)	End Point Moving Average of a
EstimatedVolume	EstimatedVolume	Estimated volume of the current bar (ProScreener command only)
EXP	EXP(a)	Mathematical Function "Exponential"
ExponentialAverage	ExponentialAverage[N](price)	Exponential Moving Average

**F - G**


---

CODE	SYNTAX	FUNCTION
FOR/TO/NEXT	FOR i=a TO b DO a NEXT	FOR loop (processes all the values with an ascending (TO) or a descending order (DOWNTO))
ForceIndex	ForceIndex(price)	Force Index indicator (determines who controls the market (buyer or seller))

**H**


---

CODE	SYNTAX	FUNCTION
High	High[N]	High of the current bar or of the n-th last bar
highest	highest[N](price)	Highest price over a number of bars to be defined
HistoricVolatility	HistoricVolatility[N](price)	Historic Volatility (or statistic volatility)
Hour	Hour[N]	Represents the hour of each bar loaded in the chart

**I - J - K**


---

CODE	SYNTAX	FUNCTION
IF/THEN/ENDIF	IF a THEN b ENDIF	Group of conditional instructions without second instruction
IF/THEN/ELSE/ENDIF	IF a THEN b ELSE c ENDIF	Group of conditional instructions
IntradayBarIndex	IntradayBarIndex[N]	Counts how many bars are displayed in one day on the whole data loaded

**L**


---

CODE	SYNTAX	FUNCTION
LinearRegression	LinearRegression[N](price)	Linear Regression indicator
LinearRegressionSlope	LinearRegressionSlope[N](price)	Slope of the Linear Regression indicator
LOG	LOG(a)	Mathematical Function "Neperian logarithm" of a
Low	Low[N]	Low of the current bar or of the n-th last bar
lowest	lowest[N](price)	Lowest price over a number of bars to be defined

**M**

CODE	SYNTAX	FUNCTION
MACD	MACD[S,L,Si](price)	Moving Average Convergence Divergence (MACD) in histogram
MACDline	MACDLine[S,L](price)	MACD line indicator
MassIndex	MassIndex[N]	Mass Index Indicator applied over N bars
MAX	MAX(a,b)	Mathematical Function "Maximum"
MedianPrice	MedianPrice	Average of the high and the low
MIN	MIN(a,b)	Mathematical Function "Minimum"
Minute	Minute	Represents the minute of each bar loaded in the chart
MOD	a MOD b	Mathematical Function "remainder of the division"
Momentum	Momentum[I]	Momentum indicator (close – close of the n-th last bar)
MoneyFlow	MoneyFlow[N](price)	MoneyFlow indicator (result between -1 and 1)
MoneyFlowIndex	MoneyFlowIndex[N]	MoneyFlow Index indicator
Month	Month[N]	Represents the month of each bar loaded in the chart

**N**

CODE	SYNTAX	FUNCTION
NEXT	See FOR/TO/NEXT	Ending Instruction of FOR loop
NOT	Not A	Logical Operator NOT

**O**

CODE	SYNTAX	FUNCTION
OBV	OBV(price)	On-Balance-Volume indicator
ONCE	ONCE VariableName = VariableValue	Introduces a definition statement which will be processed only once
Open	Open[N]	Open of the current bar or of the n-th last bar
OR	a OR b	Logical Operator OR

**P - Q**

CODE	SYNTAX	FUNCTION
PriceOscillator	PriceOscillator[S,L](price)	Percentage Price oscillator
PositiveVolumeIndex	PriceVolumeIndex(price)	Positive Volume Index indicator
PVT	PVT(price)	Price Volume Trend indicator

**R**

CODE	SYNTAX	FUNCTION
R2	R2[N](price)	R-Squared indicator (error rate of the linear regression on price)
Range	Range[N]	calculates the Range (High minus Low)
REM	REM comment	Introduces a remark (not taken into account by the code)
Repulse	Repulse[N](price)	Repulse indicator (measure the buyers and sellers force for each candlestick)
RETURN	RETURN Result	Instruction returning the result
ROC	ROC[N](price)	Price Rate of Change indicator
RSI	RSI[N](price)	Relative Strength Index indicator
ROUND	ROUND(a)	Mathematical Function "Round a to the nearest whole number"

**S**

CODE	SYNTAX	FUNCTION
SAR	SAR[At,St,Lim]	Parabolic SAR indicator
SARatdmf	SARatdmf[At,St,Lim](price)	Smoothed Parabolic SAR indicator
SCREENER	SCREENER[c](price)	Display results of the ProScreener (ProScreener command only)
SIN	SIN(a)	Mathematical Function "Sine"
SGN	SGN(a)	Mathematical Function "Sign of" a (it is positive or negative)
SMI	SMI[N,SS,DS](price)	Stochastic Momentum Index indicator
SmoothedStochastic	SmoothedStochastic[N,K](price)	Smoothed Stochastic
SORT BY	SORT BY	Filters or sorts the results of the screener (ProScreener command only)
SQUARE	SQUARE(a)	Mathematical Function "a Squared"

<code>SQRT</code>	<code>SQRT(a)</code>	Mathematical Function "Squared Root" of a
<code>STD</code>	<code>STD[N](price)</code>	Statistical Function "Standard Deviation"
<code>STE</code>	<code>STE[N](price)</code>	Statistical Function "Standard Error"
<code>Stochastic</code>	<code>Stochastic[N,K](price)</code>	%K Line of the Stochastic indicator
<code>summation</code>	<code>summation[N](price)</code>	Sums a certain price over the N last candlesticks
<code>Supertrend</code>	<code>Supertrend[STF,N]</code>	Super Trend indicator

## T

CODE	SYNTAX	FUNCTION
<code>TAN</code>	<code>TAN(a)</code>	Mathematical Function "Tangent" of a
<code>TEMA</code>	<code>TEMA[N](price)</code>	Triple Exponential Moving Average
<code>THEN</code>	See IF/THEN/ELSE/ENDIF	Instruction following the first condition of "IF"
<code>Time</code>	<code>Time[N]</code>	Represents the time of each bar loaded in the chart
<code>TimeSeriesAverage</code>	<code>TimeSeriesAverage[N](price)</code>	Temporal series moving average
<code>TIMEFRAME</code>	<code>TIMEFRAME(default)</code>	Allows you to write conditions for your ProScreener in different timeframes (ProScreener command only)
<code>TO</code>	See FOR/TO/NEXT	Directional Instruction in the "FOR" loop
<code>Today</code>	<code>Today[N]</code>	Date of the bar n-periods before the current bar
<code>TotalPrice</code>	<code>TotalPrice[N]</code>	$(Close + Open + High + Low) / 4$
<code>TR</code>	<code>TR(price)</code>	True Range indicator
<code>TriangularAverage</code>	<code>TriangularAverage[N](price)</code>	Triangular Moving Average
<code>TRIX</code>	<code>TRIX[N](price)</code>	Triple Smoothed Exponential Moving Average
<code>TypicalPrice</code>	<code>TypicalPrice[N]</code>	Represents the Typical Price (Average of the High, Low and Close)

## U

CODE	SYNTAX	FUNCTION
<code>Undefined</code>	<code>a = Undefined</code>	Sets a the value of a variable to undefined

**V**

CODE	SYNTAX	FUNCTION
Variation	Variation(price)	Difference between the close of the last bar and the close of the current bar in %
Volatility	Volatility[S, L]	Chaikin volatility
Volume	Volume[N]	Volume indicator
VolumeOscillator	VolumeOscillator[S,L]	Volume Oscillator
VolumeROC	VolumeROC[N]	Volume of the Price Rate Of Change

**W**

CODE	SYNTAX	FUNCTION
WeightedAverage	WeightedAverage[N](price)	Represents the Weighted Moving Average
WeightedClose	WeightedClose[N]	Average of (2 * Close), (1 * High) and (1 * Low)
WEND	See WHILE/DO/WEND	Ending Instruction of WHILE loop
WHILE/DO/WEND	WHILE (condition) DO (action) WEND	WHILE loop
WilderAverage	WilderAverage[N](price)	Represents Wilder Moving Average
Williams	Williams[N](close)	%R de Williams indicator
WilliamsAccumDistr	WilliamsAccumDistr(price)	Accumulation/Distribution of Williams Indicator

**X**

CODE	SYNTAX	FUNCTION
XOR	a XOR b	Logical Operator eXclusive OR

**Y**

CODE	SYNTAX	FUNCTION
Year	Year[N]	Year of the bar n periods before the current bar
Yesterday	Yesterday[N]	Date of the day preceeding the bar n periods before the current bar

**Z**


---

CODE	SYNTAX	FUNCTION
<code>ZigZag</code>	<code>ZigZag[Zr](price)</code>	Represents the Zig-Zag indicator introduced in the Elliott waves theory
<code>ZigZagPoint</code>	<code>ZigZagPoint[Zp](price)</code>	Represents the Zig-Zag indicator in the Elliott waves theory calculated on Zp points

**Other**


---

CODE	FUNCTION	CODE	FUNCTION
<code>+</code>	Addition Operator	<code>&lt;</code>	Strict Inferiority Operator
<code>-</code>	Substraction Operator	<code>&gt;</code>	Strict Superiority Operator
<code>*</code>	Multiplication Operator	<code>&lt;=</code>	Inferiority Operator
<code>/</code>	Division Operator	<code>&gt;=</code>	Superiority Operator
<code>=</code>	Equality Operator	<code>//</code>	Introduces a commentary line
<code>&lt;&gt;</code>	Difference Operator		

# ProRealTime SOFTWARE

