



# ***NanoTrader V4***

## **Release Notes**

Juli, 2021



## Content

1	Willkommen zu NanoTrader V4 .....	2
2	Erscheinungsbild .....	3
3	Der neue Farb-Manager .....	6
4	Neuer „Sentimentor Visualisierung“-Dialog .....	7
5	Verschiedene Erweiterungen der Oberfläche .....	7
6	Neue Express Umgebung .....	9
6.1	Neuer Express Editor .....	9
6.2	Neuer Express Debugger .....	13
6.3	Neue Express Funktionalitäten .....	18

## 1 Willkommen zu NanoTrader V4

Mit NanoTrader V4 veröffentlicht Fipertec eine brandneue Version seiner preisgekrönten Handelsplattform. NanoTrader V4 präsentiert sich in einem frischen Erscheinungsbild und bietet eine Vielzahl neuer Funktionalitäten, basierend auf seiner grundsoliden und innovativen Technologie. Hier sind einige Highlights von NanoTrader V4:

- **uneingeschränkte Abwärtskompatibilität**  
NanoTrader V4 ist vollständig abwärtskompatibel mit allen Vorgängerversionen. Es ist keine einzige Änderung in Ihren Studien und Seiten erforderlich, um die Vorteile der neuen Version zu nutzen.
- **64 Bit**  
NanoTrader V4 ist in 32- und 64-Bit-Versionen verfügbar. Das Installationsprogramm ermittelt automatisch, welche Version zu installieren ist. Auch ein Upgrade von 32 auf 64 Bit erfordert keinerlei Änderungen Ihrer persönlichen Einstellungen.
- **Erscheinungsbild**  
Sie können aus vier vordefinierten Designs wählen, die das Erscheinungsbild des Hauptfensters und der Dialoge bestimmen. Darunter ist eine "Nacht"-Variante, die nur dunkle Farben verwendet.
- **Duale Farben**  
Um die Verwendung von dunklen Chart-Hintergründen zu vereinfachen, wurde das gesamte Farbmanagement neu gestaltet. Wenn für einen dunklen Hintergrund keine Farbe explizit angegeben wurde, adaptiert NanoTrader die aktuelle Farbe automatisch so, dass sie auch nach Umschalten auf einen dunklen Hintergrund gut aussieht.
- **Neue Express Umgebung**  
Die Express-Umgebung verfügt jetzt über einen umfassenden Editor. Der Express-Editor ist jetzt nicht mehr blockierend, so dass Sie so viele Editoren gleichzeitig öffnen können, wie Sie möchten. Außerdem wurde

ein Debugger hinzugefügt, der es ermöglicht, Code-Probleme leicht zu erkennen. Eine innovative Funktion erlaubt es, Breakpoints direkt im Chart zu setzen.

- **Express Sprach-Erweiterungen**  
Neue Funktionen in Express ermöglichen es, Haltepunkte zu definieren. Auch die neuen dualen Farben werden unterstützt. Außerdem kann die Deckkraft für gefüllte Bänder, Highlight() und Annotation() festgelegt werden.
- **Weitere Verbesserungen der Benutzerfreundlichkeit**  
Für die Bereitstellung des neuen Erscheinungsbildes wurden alle Dialoge von NanoTrader überarbeitet. Im Zuge dessen wurden viele kleine Verbesserungen implementiert, z.B. neue Icons, verschönerte Infobox, reaktionsschnellere Suche, verbesserte Kontextmenüs, neue Hotkeys, neuer Farb-Manager, neuer Optionen-Dialog und vieles mehr.

Dieses Dokument hebt die neuen Funktionen von NanoTrader V4 hervor. Eine detaillierte Beschreibung finden Sie in der Dokumentation, die über das Hilfe-Menü von NanoTrader verfügbar ist.

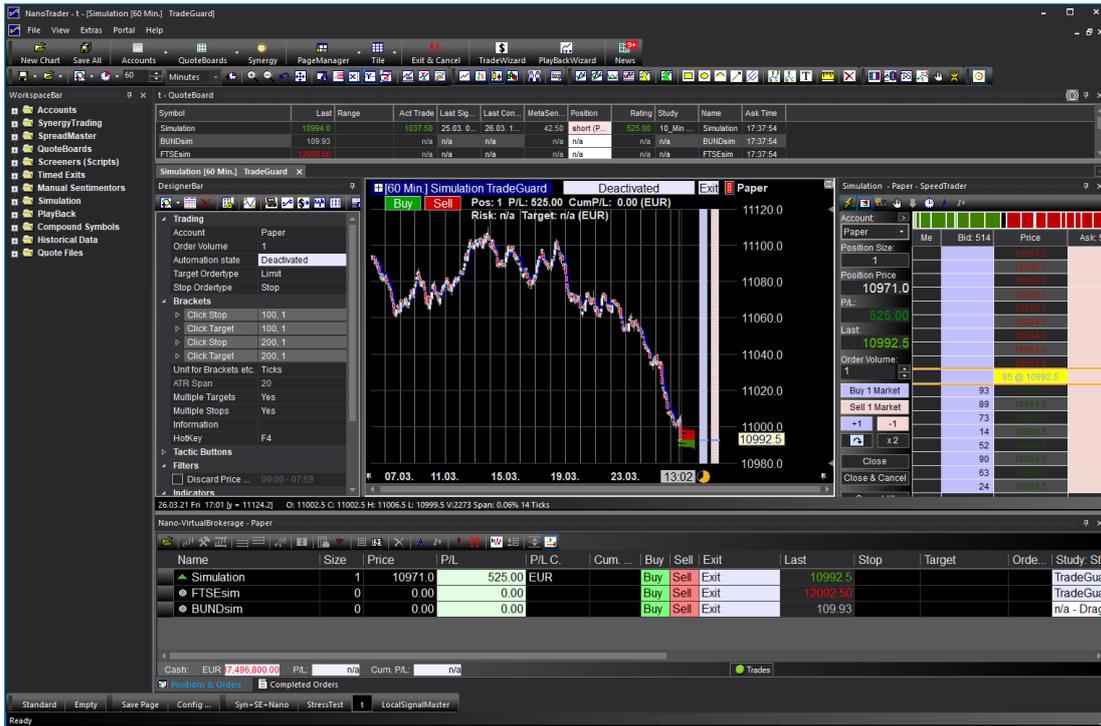
## 2 Erscheinungsbild

Das Erscheinungsbild bestimmt die Farben und den Zeichenstil der Fenster und Dialoge von NanoTrader. Er wird durch ein Farbschema ergänzt, das alle Farben der von NanoTrader gezeichneten Elemente definiert, z. B. die Farben im SpeedTrader, Charts, Kursleisten, Preis-Schiebern usw.

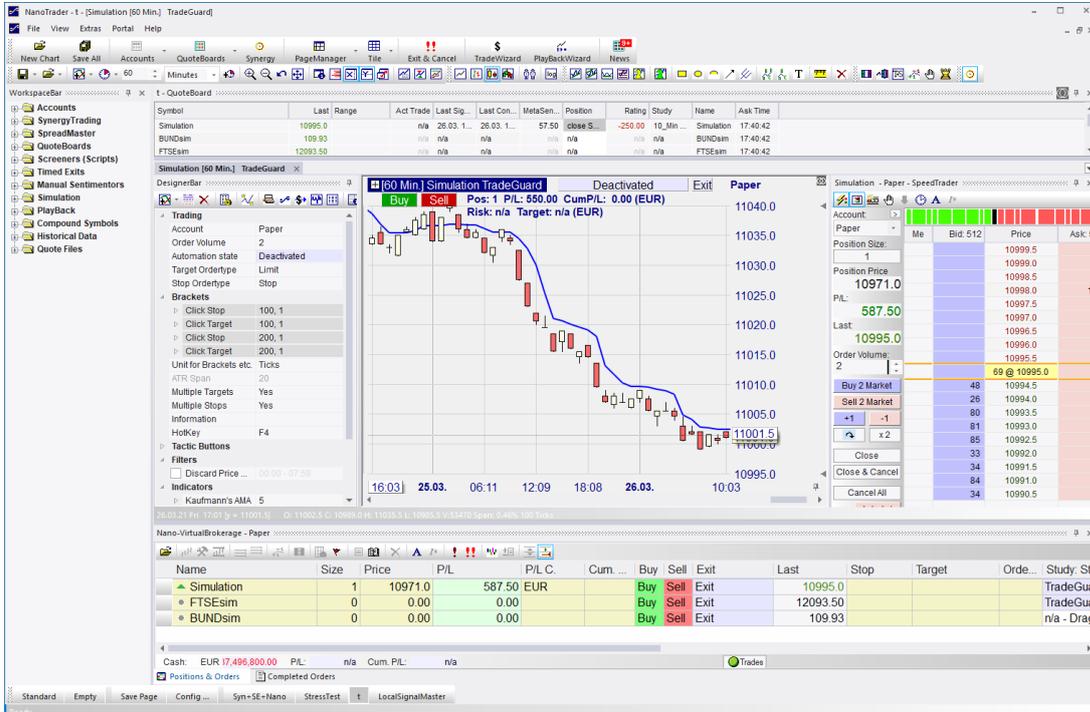
NanoTrader V4 wird mit vier vordefinierten System-Farbschemata ausgeliefert. Um Ihre persönlichen Schemata zu erstellen, wählen Sie das Systemschema aus, das Ihrem Geschmack am besten entspricht und passen es nach Ihren Wünschen an.

Hier sehen Sie einige Screenshots der von Fipertec bereitgestellten Schemata:

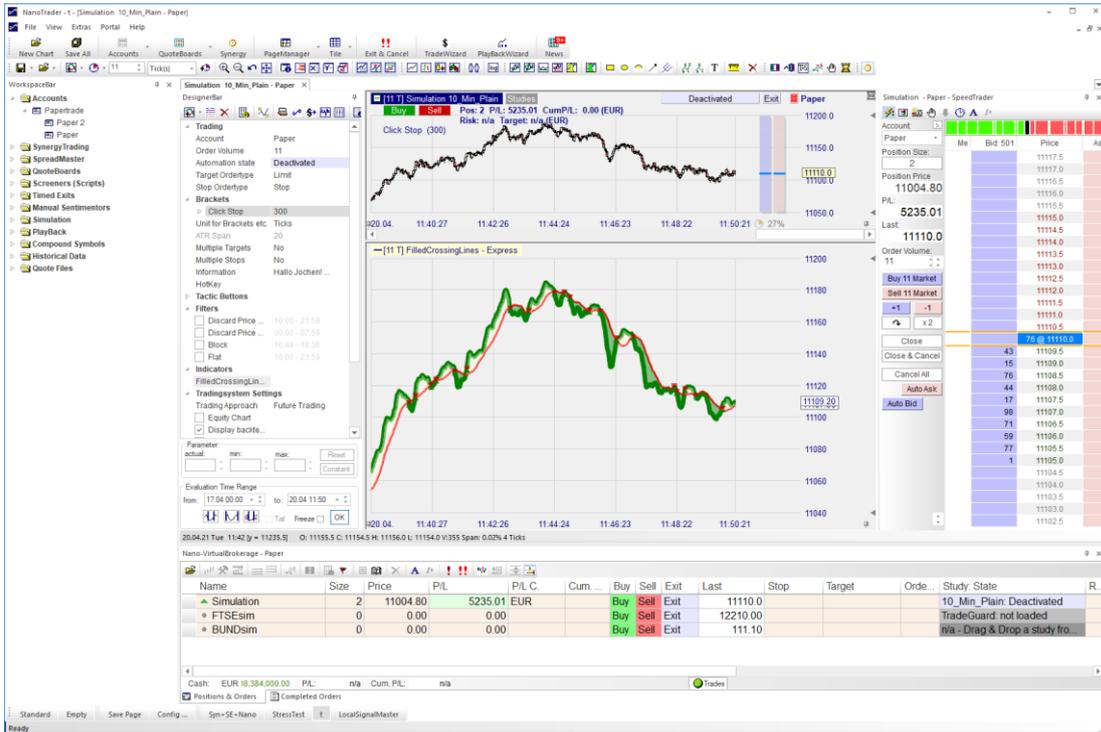
## Schema Night



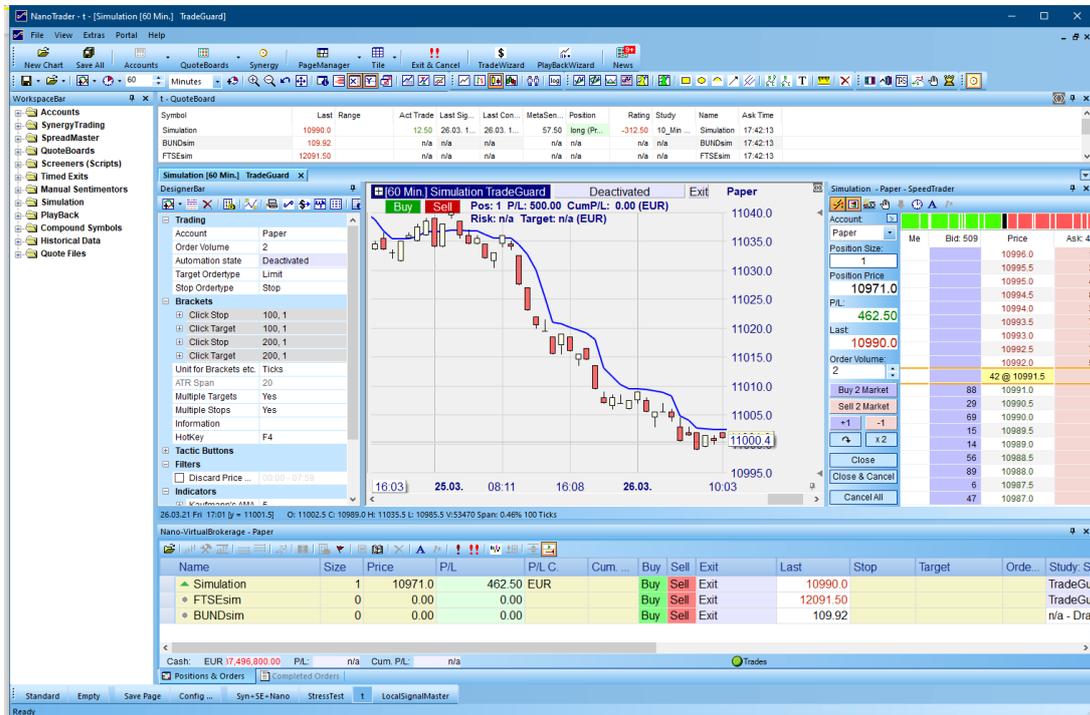
## Schema Silver



## Schema White

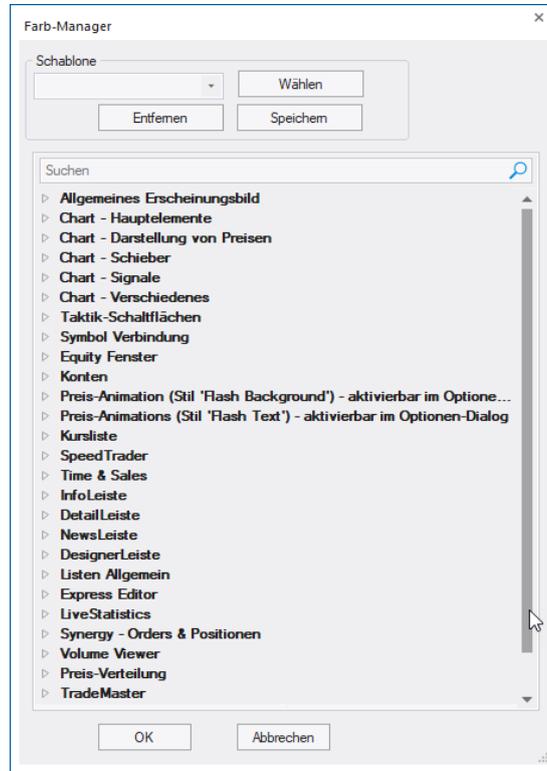


## Schema Classic

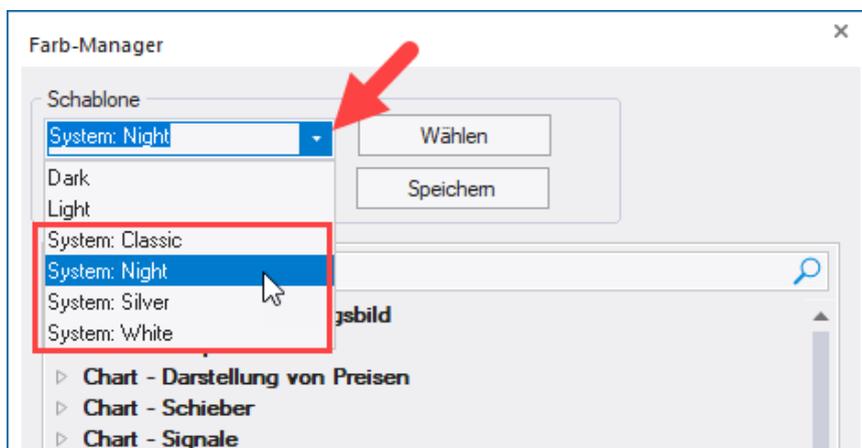


### 3 Der neue Farb-Manager

Der Farb-Manager-Dialog wurde komplett überarbeitet und ermöglicht die komfortable Definition persönlicher Farbschemata:



- Die Farben wurden in Kategorien eingeteilt. Die Kategorien sind nach Relevanz sortiert.
- Das Erscheinungsbild kann unter "Allgemeines Erscheinungsbild" festgelegt werden. Beachten Sie, dass Sie in der Regel zunächst ein System-Farbschema im Bereich "Schablone" auswählen und von dort aus Änderungen vornehmen.



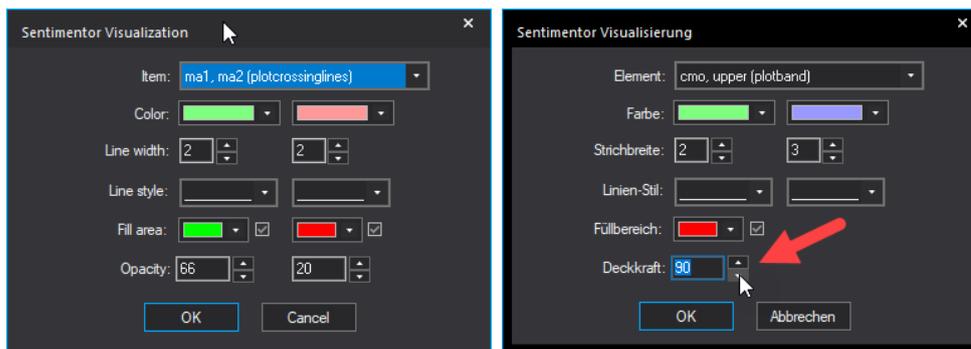
- In dem "Suchen"-Feld kann nach Farben gesucht werden.

- Jede Änderung einer Farbe ist sofort wirksam, ohne dass NanoTrader neu gestartet werden muss.
- Der neue Farb-Manager ist nicht blockierend. Er kann somit geöffnet bleiben, während Sie mit NanoTrader arbeiten und die Änderungen testen.
- Der neue Farb-Manager unterstützt das Kopieren und Einfügen von Farben mit dem Windows-Standard Strg-C und Strg -V sowie das Rückgängigmachen und Wiederherstellen mit dem Windows-Standard Strg -Z und Strg -Y.

#### 4 Neuer „Sentimentor Visualisierung“-Dialog

Die Visualisierungsdialoge für Indikatoren wurden aktualisiert, um die Angabe der Deckkraft von Füllfarben zu ermöglichen.

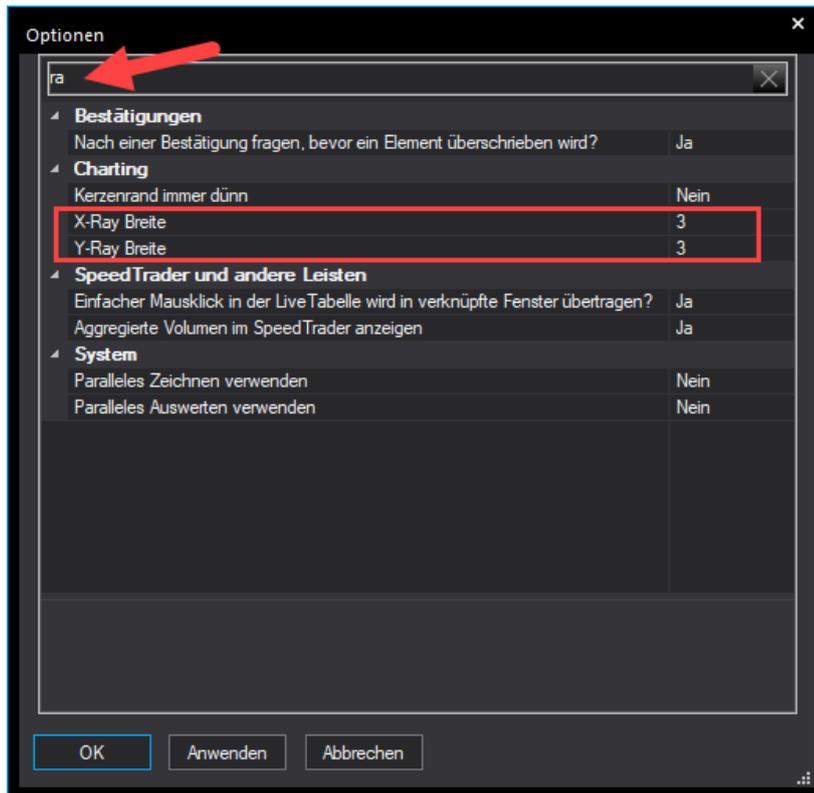
Außerdem wird jede Änderung eines beliebigen Wertes sofort visualisiert. Daher ist die Schaltfläche "Anwenden" verschwunden:



Der Visualisierungsdialog unterstützt die neuen dualen Farben, d. h. die geänderte Farbe wird nur für den gerade aktiven Chart-Hintergrund aktiv.

#### 5 Verschiedene Erweiterungen der Oberfläche

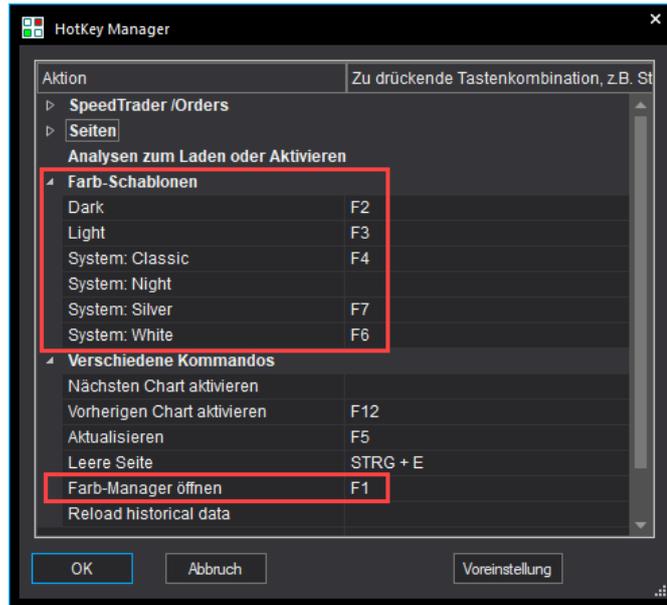
- Der neue Optionen-Dialog hat ein Suchfeld und ist in der Größe veränderbar.
- Die Breite der X/Y-Rays kann im Optionen-Dialog festgelegt werden:



- Die X/Y-Rays erscheinen nun genau in der ihnen zugewiesenen Farbe.
- Die Zeichnung des Kreisdiagramms für die Perioden-Komplettierung erfolgt jetzt in hoher Auflösung. Die "Prozent"-Anzeige, falls aktiviert, wird neben dem Kreisdiagramm angezeigt:



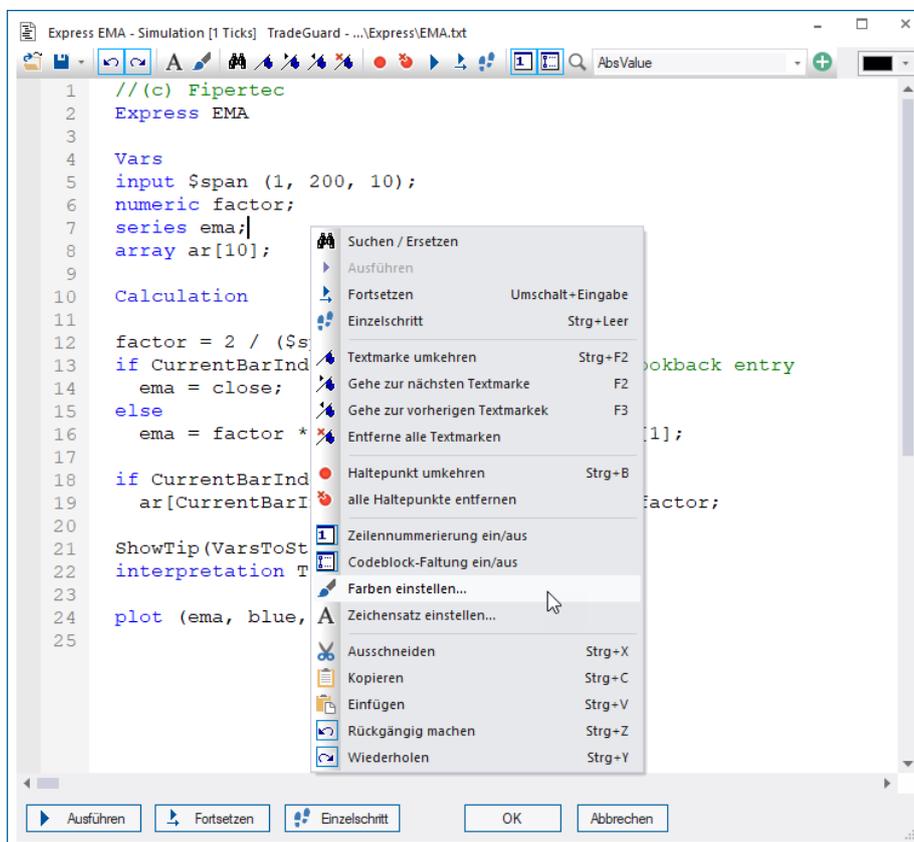
- Verbesserte "Trefferprüfung" für linienbasierte Zeichenwerkzeuge, d.h. Linien können einfacher mit der Maus selektiert werden.
- Preisboxen werden genauer und ohne Schatten gezeichnet.
  - Neue Hotkey-Befehle zum Starten des Farb-Managers und zum Umschalten auf eines der Farbschemata:



## 6 Neue Express Umgebung

### 6.1 Neuer Express Editor

Aufgrund der gestiegenen Nachfrage nach der Programmierung von komplexem Express-Code wurde der ursprünglich minimalistische Express-Editor komplett überarbeitet:



Hier sind einige Funktionen des neuen Express-Editors:

- **Nicht-Blockierend**

Der Express-Editor ist nicht mehr blockierend, d.h., Sie können mehrere Editoren gleichzeitig öffnen und in den MasterChart hineinzoomen, um verschiedene Bereiche des Charts zu untersuchen.

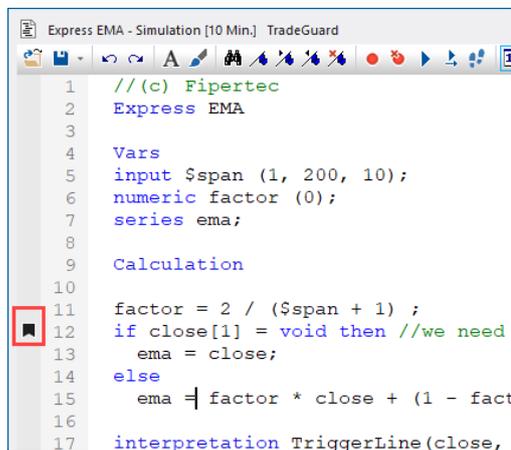
Es ist auch möglich, die Parameter des aktuell bearbeiteten Express-Sentimentors in der DesignerLeiste zu ändern und die Aggregation des MasterCharts oder des Sentimentors zu ändern.

- **Zeilennummern**

Zeilennummern können eingeblendet werden.

- **Textmarken**

Setzen Sie Lesezeichen an wichtigen Zeilen und navigieren Sie schnell zwischen ihnen:



```

Express: EMA - Simulation [10 Min.] TradeGuard
1 // (c) Fipertec
2 Express EMA
3
4 Vars
5 input $span (1, 200, 10);
6 numeric factor (0);
7 series ema;
8
9 Calculation
10
11 factor = 2 / ($span + 1) ;
12 if close[1] = void then //we need
13     ema = close;
14 else
15     ema = factor * close + (1 - fact
16
17 interpretation TriqgerLine(close,
  
```

- **Codeblock-Faltung**

Ermöglicht das Ein/Ausblenden von Codeblöcken, die mit begin/end eingeschlossen sind:

```

6 series cmo, up, down;
7 series upper;
8 numeric sumUp(0), sumDown(0);
9 numeric move;
10
11 Calculation
12 Break();
13 if CurrentBarIndex() > 0 then
14     move = close - close[1];
15
16 if CurrentBarIndex() < $span then
17     //correct the sums:
18     //Break();
19     sumUp = sumUp - up[$span];
20     sumDown = sumDown - down[$span];
21
22     //BreakIf (time > 8_5_2021);
23     if (CurrentBarIndex() > (FinalBarIndex() - 10)) then
24         ShowTip(VarsToString(""));
25     //else
26     // Highlight("textAbove:" + VarsToString(""), "blue");
27
28 if move > 0 then
29     begin
30         sumUp = sumUp + move;
31         sumDown = sumDown - move;
32         cmo = void; //no data
33     end
34
35 else
36     begin
37         sumUp = sumUp - move;
38         sumDown = sumDown + move;
39         cmo = void; //no data
40     end
41
42 end
43

```

- **Haltepunkte**

Setzen Sie Haltepunkte und die Codeausführung wird in diesen Zeilen angehalten.

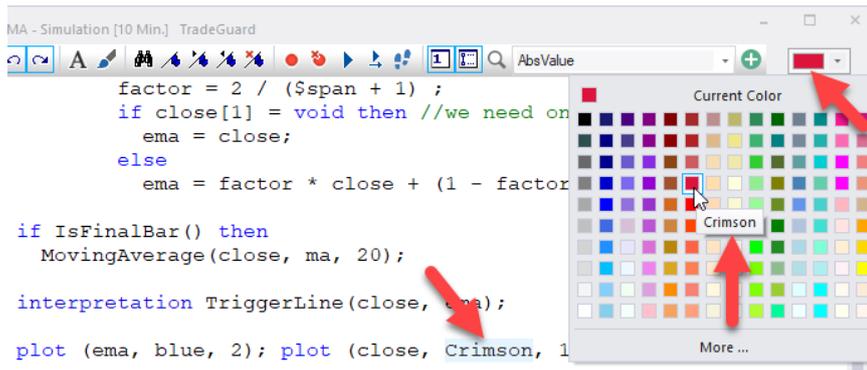
```

1 // (c) Fipertec
2 Express EMA
3
4 Vars
5 input $span (1,
6 numeric factor (
7 series ema;
8
9 Calculation
10
11 factor = 2 / ($s
12 if close[1] = vo
13     ema = close;
14 else
15 | ema = factor *
16
17 interpretation T
18
19 plot (ema, blue,

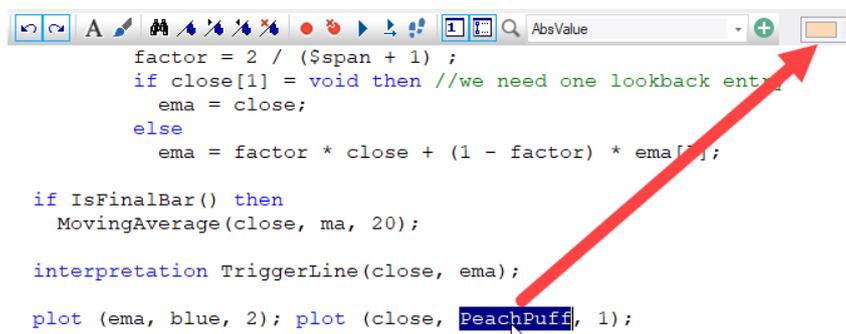
```

- **Farbwähler & Anzeiger**

Das Festlegen ansprechender Farben für geplottete Elemente ist wesentlich, um das Beste aus Ihrem Express-Code herauszuholen. Um eine Farbe auszuwählen, öffnen Sie das Dropdown-Menü und wählen eine Farbe aus. Der Name der gewählten Farbe bzw. ihr Farb-Code wird an der aktuellen Cursorposition eingefügt:

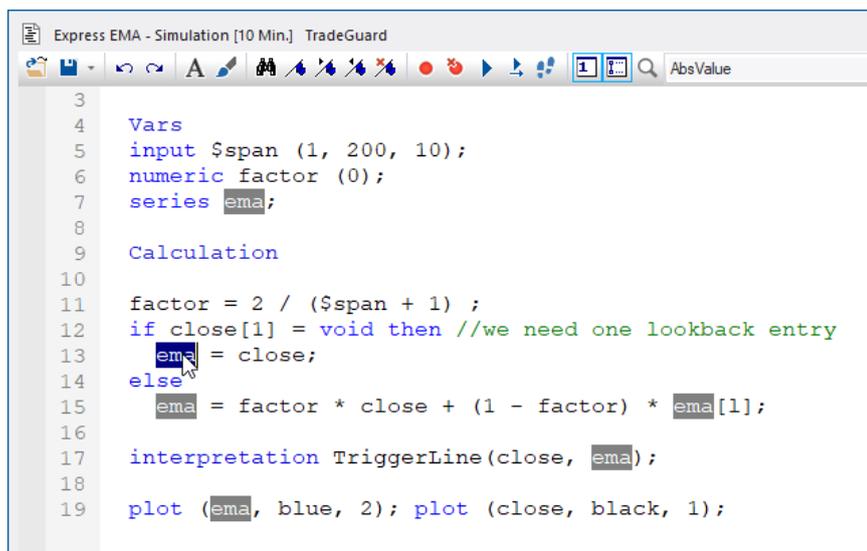


Um sich die Farbe zu einem einem Farbcode anzeigen zu lassen, genügt es, diesen per Doppelklick zu selektieren:



- **Hervorheben identischer Wörter**

Wenn Sie ein Wort doppelklicken, werden alle identischen Wörter im Text markiert. Dies ist sehr hilfreich, um schnell zu erkennen, wo eine Variable oder Funktion verwendet wird:



- **Einrücken von Blöcken**

Um einen Zeilenblock einzurücken, markieren Sie ihn mit der Maus und drücken dann die Tabulatortaste. Drücken Sie Shift-Tab, um die Einrückung aufzuheben:

```

9 Calculation
10
11 factor = 2 / ($span + 1) ;
12 if close[1] = void then //we need one lookback entry
13     ema = close;
14 else
15     ema = factor * close + (1 - factor) * ema[1];

```

- **Funktions-Signaturen anzeigen**

Bewegen Sie die Maus über eine Funktion, um deren Signatur zu sehen:

```

15     ema = factor * close + (1 - factor) * ema
16
17     if IsFinalBar() then
18         MovingAverage(close, ma, 20);
19         void MovingAverage(series source, series target, int span)
20     interpretation TriggerLine(close, ema);
21

```

- **Mehrzeilige Kommentare**

Mehrzeilige Kommentare mit { und } werden jetzt als Kommentare eingefärbt:

```

32     plotcandles(o,c,h,l);
33
34     {
35         plotline (11050, "IndianRed", 2);
36         plot (ma1, "black", 2);
37         plot (ma3, red, 2);
38     }

```

## 6.2 Neuer Express Debugger

Die Express-Umgebung ist nun mit einem sogenannten "Debugger" ausgestattet. Ein Debugger ermöglicht es, die Codeausführung an bestimmten Haltepunkten im Code anzuhalten und die Variablen und den Programmablauf zu untersuchen. Dadurch lassen sich Fehler leicht erkennen und beheben.

### 6.2.1 Haltepunkte setzen

Der schnellste Weg, einen Haltepunkt zu setzen, ist ein Klick in den Marker-Bereich auf der linken Seite des Editors:

```

1 // (c) Fipertec
2 Express EMA
3
4 Vars
5 input $span (1, 200, 10);
6 numeric factor;
7 series ema, ma;
8
9 Calculation
10
11 factor = 2 / ($span + 1) ;
12 if CurrentBarIndex()=0 then //we need one lookback entry
13     ema = close;
14 else
15     ema = factor * close + (1 - factor) * ema[1];
16
17 ShowTip(VarsToString("close,factor"));
18

```

Alternativ kann ein Haltepunkt in der aktuellen Zeile über die Symbolleiste, das Kontextmenü oder durch Drücken von Strg-B gesetzt werden.

Es können beliebig viele Haltepunkte gesetzt werden.

Wenn ein Haltepunkt in einer Zeile gesetzt wird, die keinen unterbrechbaren Code enthält, wird der Haltepunkt automatisch in eine solche verschoben, sobald die Programmausführung beginnt oder fortgesetzt wird.

Sobald während der Programmausführung ein Haltepunkt auftritt, wird die Ausführung gestoppt und die Zeile, in der die Unterbrechung aufgetreten ist, wird hervorgehoben:

```

1 // (c) Fipertec
2 Express EMA
3
4 Vars
5 input $span (1, 200, 10);
6 numeric factor;
7 series ema, ma;
8
9 Calculation
10
11 factor = 2 / ($span + 1) ;
12 if CurrentBarIndex()=0 then //we need one lookback entry
13     ema = close;
14 else
15     ema = factor * close + (1 - factor) * ema[1];
16
17 ShowTip(VarsToString("close,factor"));
18

```

Die Farbe für die Hervorhebung kann im Farb-Manager eingestellt werden.

Beachten Sie, dass im obigen Beispiel die else-Klausel ausgeführt wurde. Ansonsten wäre der Haltepunkt ignoriert worden.

Das Programm wird angehalten, bevor der Code auf der Haltepunktzeile ausgeführt wird.

### Haltepunkte im Editor entfernen

Um einen einzelnen Haltepunkt zu entfernen, klicken Sie ihn im Markierungsbereich an oder platzieren Sie den Cursor in der Zeile daneben und verwenden Sie die Schaltfläche  in der Toolbar, das Kontextmenü oder Strg-B.

Um alle Haltepunkte zu entfernen, klicken Sie  in der Toolbar oder im Kontextmenü.

## Haltepunkte durch Funktionsaufrufe

Manchmal möchte man die Programm-Ausführung nur unter bestimmten Bedingungen unterbrechen. Um das zu erreichen, stehen die folgenden drei Express-Funktionen zur Verfügung. Diese Funktionen sind nur aktiv, wenn der Code im Editor geöffnet ist. Andernfalls werden, werden sie ignoriert.

- **Break()**

Unterbricht die Ausführung an der Stelle, wo der Funktionsaufruf geschieht:

```

9 Calculation
10
11 factor = 2 / ($span + 1) ;
12 if CurrentBarIndex()=0 then //we need one lookback entry
13     ema = close;
14 else
15     ema = factor * close + (1 - factor) * ema[1];
16 Break();
17 ShowTip(VarsToString("close,factor"));

```

- **BreakIf (boolescher Ausdruck)**

Unterbricht die Ausführung, falls der boolesche Ausdruck `true` ergibt:

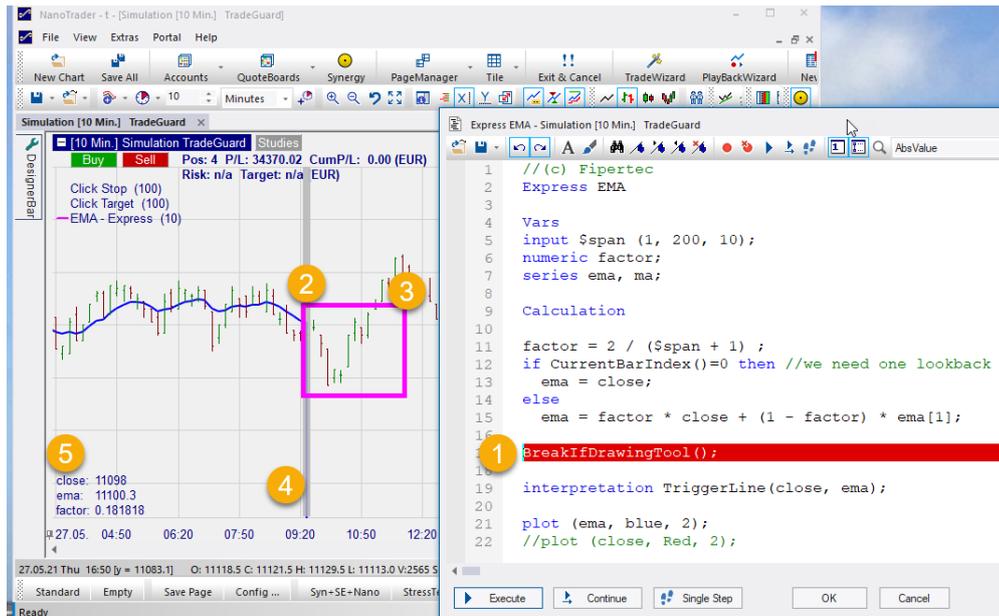
```

9 Calculation
10
11 factor = 2 / ($span + 1) ;
12 if CurrentBarIndex()=0 then //we need one lookback entry
13     ema = close;
14 else
15     ema = factor * close + (1 - factor) * ema[1];
16 BreakIf((close > open) and (CurrentBarIndex() > 5));
17 ShowTip(VarsToString("close,factor"));
18

```

- **BreakIfDrawingTool()**

Beim Ausführen dieser Funktion prüft NanoTrader, ob *in der aktuellen Periode* im MasterChart ein Zeichenwerkzeug beginnt oder endet. Im untenstehenden Beispiel wurde ein Rechteck um einen "interessanten" Bereich gezeichnet. NanoTrader wird am ersten und letzten abgedeckten Balken die Programm-Ausführung unterbrechen:



(Weitere Erklärungen finden Sie im nächsten Abschnitt.)

## 6.2.2 Variablen während eines Programm-Halts inspizieren

### Im MasterChart

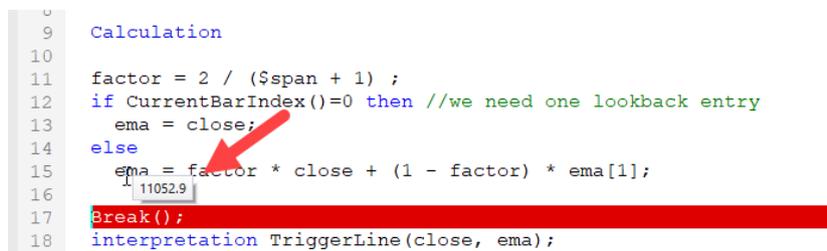
Sobald ein Haltepunkt auftritt, zeigt NanoTrader einen grauen vertikalen Balken im MasterChart an, um die Periode hervorzuheben, die gerade bearbeitet wird.

Siehe **4** im obigen Beispiel. Wenn die Periode außerhalb des aktuellen Zooms liegt, wird dieser automatisch angepasst.

Alle im Code verwendeten Variablen sowie ihre aktuellen Werte werden im unteren Teil des MasterCharts angezeigt **5**. "Verwendet" bedeutet, dass die Variable oder Serie nicht nur definiert ist, sondern auch im Code referenziert wird.

### Im Editor

Während die Programmausführung angehalten ist, können Sie mit der Maus auf eine Variable oder Serie zeigen und ihr aktueller Wert wird in einem Pop-up-Fenster angezeigt:



Wenn auf ein Array gezeigt wird, wird dessen gesamter Inhalt angezeigt:

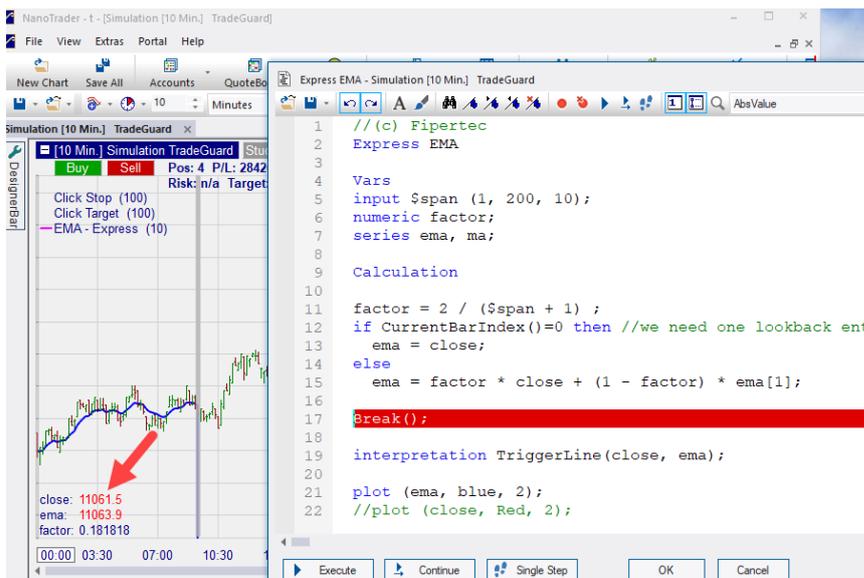
```

4  Vars
5  input $span (1, 200, 10);
6  numeric factor;
7  series ema, ma;
8  array ar[10];
9
10 Calculation
11
12 factor = 2 / ($span + 1) ;
13 if CurrentBarIndex()==0 then //we need one lookback
14     ema = close;
15 else
16     ema = factor * close + (1 - factor) * ema[1];
17
18 if CurrentBarIndex() < 10 then
19     ar[CurrentBarIndex()] = (open - close) / factor;
20
21 Break();
22 int TriggerLine(close, ema);
23
24 plot (ema, blue, 2);

```

### 6.2.3 Hervorhebung von geänderten Variablen

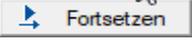
Der Wert einer Variablen wird im Chart rot dargestellt, wenn sich ihr Wert im Vergleich zum Wert beim vorherigen Haltepunkt geändert hat:



### 6.2.4 Fortsetzen der Code-Ausführung von einem Haltepunkt aus

Es gibt drei Möglichkeiten mit der Code-Ausführung fortzufahren, nachdem ein Haltepunkt ausgeführt wurde:

- Ausführen  **Ausführen**  
Wenn Sie auf die Schaltfläche "Ausführen" klicken, wird der Code *erneut* ausgeführt, d. h. die Ausführung beginnt mit der ersten Periode und der ersten Anweisung. Nachdem der Code geändert wurde, sollten Sie immer "Ausführen" verwenden, um das Skript erneut auszuführen.

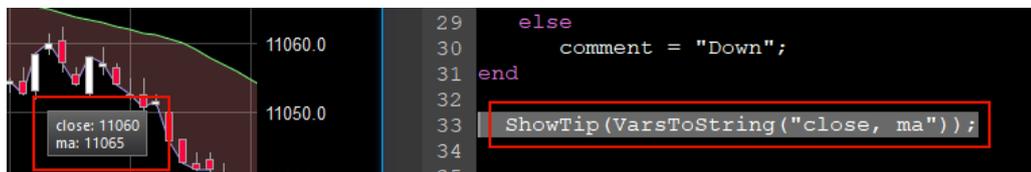
- Fortsetzen    
Die Programmausführung wird an der Stelle fortgesetzt, wo sie angehalten wurde.
- Einzelschritt    
Die Programmausführung wird an der Stelle fortgesetzt, wo sie angehalten wurde, unterbricht aber automatisch in der nächsten Zeile. Auf diese Weise lässt sich der Programmablauf leicht verfolgen, ohne dass in jeder Zeile explizit ein Haltepunkt gesetzt werden müsste. Desweiteren lassen sich so Wertänderungen der Variablen einfach nachvollziehen.

## 6.3 Neue Express Funktionalitäten

### 6.3.1 Neue Express Funktionen

Die folgenden neuen Express-Funktionen wurden hinzugefügt. Eine ausführliche Beschreibung finden Sie im Express-Handbuch.

- **Break()**  
Unterbricht die Ausführung an der Stelle, wo der Funktionsaufruf geschieht.
- **BreakIf(boolean expression)**  
Unterbricht die Ausführung, falls der boolesche Ausdruck `true` ergibt
- **BreakIfDrawingTool()**  
Unterbricht die Ausführung, wenn ein Zeichenwerkzeug im zugehörigen MasterChart beginnt oder endet.
- **VarsToString("varname,varname,...")**  
Diese Routine ermöglicht es, auf einfache Weise Variablen und ihren Inhalt in `Highlight()`, `Annotate()` oder `ShowTip()` zu verwenden. `VarsToString()` erwartet eine kommaseparierte Liste von Variablennamen.  
Beispiel.:  
`ShowTip(VarsToString("close,ma"));`



### 6.3.2 Importieren von Arrays aus einem Sentimentor oder einer Studie

Es ist jetzt möglich, ein Array aus einem anderen Sentimentor zu importieren.

Beispiele:

```
Express Array_Exporter
```

```
Vars
numeric i;
array levels[500];
Calculation
...
levels[i] = close - open;
...
```

Arrays können mit den gleichen Namenskonventionen importiert werden, wie sie für den Import einer Serie gelten:

```
Express Array_Importer
Vars
array importedArray[ArrayExporterExpress.levels];
series test;
Calculation
if CurrentBarIndex() < GetArraySize(importedArray) then
test = importedArray[CurrentBarIndex()];
interpretation begin end
plot (test, primary, 2);
```

Mit Hilfe des Study-Sentimentors ist es möglich, ein Array aus einem Sentimentor zu importieren, der sich in einer anderen Analyse befindet:

```
array importedArray[study.ArrayExporterExpress.levels];
```

### 6.3.3 Unterstützung dualer Farben in Express

#### Allgemeines Prinzip

Eine duale Farbe besteht immer aus der Farbangabe für einen hellen Chart-Hintergrund, optional gefolgt von einer Farbangabe für einen dunklen Chart-Hintergrund. Das Tilde-Zeichen ~ trennt beide Angaben:

Beispiele:

```
plot (close, "black", 1);
```

Verwendet Schwarz auf einem hellen Chart-Hintergrund und die automatisch angepasste Farbe auf einem dunklen Chart-Hintergrund.

```
plot (close, "black~white", 1);
```

Verwendet Schwarz auf einem hellen Hintergrund und Weiß auf einem dunklen.

```
plot (close, "128,128,255~yellow", 1);
```

Farbangabe für hellen Hintergrund in RGB-Notation.

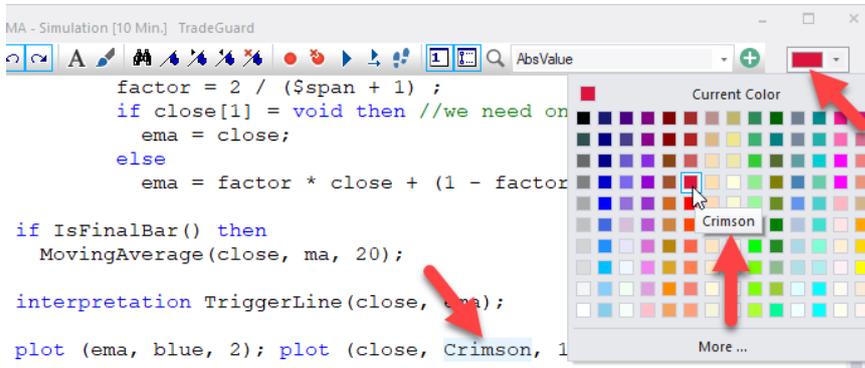
```
plot (close, "1325324~0,0,255", 1);
```

Verwendet den Integer-Wert der RGB-Notation für den hellen Hintergrund.

Wenn keine Farbe für einen dunklen Hintergrund angegeben wird, bestimmt NanoTrader eine Farbe auf der Grundlage der Farbe für den hellen

Hintergrund, indem es deren Leuchtkraft invertiert. Aus einem "Dunkelblau auf hellem Hintergrund" wird dann ein "Hellblau auf dunklem Hintergrund".

Ein gültiger Farbname kann am einfachsten über den Farbwähler in der Toolbar in den Code eingefügt werden:



## Logische Farben

Zusätzlich zu den Farbnamen kann ein Satz von *logischen Farben* verwendet werden. Logische Farben sind bereits für helle und dunkle Hintergründe vordefiniert. Sie können bequem verwendet werden, um auf gut aussehende Farben für typische Anwendungsfälle zurückzugreifen. Die Verwendung der gleichen Farben in ähnlichen Szenarien erleichtert das "Lesen" und Verstehen von Sentimentor-Grafiken.

Die logischen Farbnamen lauten:

- Primary Die primäre Serie in einem Chart
- Secondary Die sekundäre Serie, bspw. bei CrossingMA
- UpperBand
- LowerBand
- FillBand Die Standard Füll-Farbe
- UpperThreshold
- LowerThreshold
- Threshold

Beispiele:

```
plot (close, "primary", 1);
```

```
plot (ma, "secondary", 1);
```

Hinweis: Eine logische Farbe definiert implizit Farben für einen hellen und dunklen Hintergrund. Daher ist eine Angabe wie "primary~red" ungültig und wird vom Express-Compiler nicht akzeptiert.

## Angabe der Deckkraft für Füllfarben, Highlight() und Annotate()

Eine Farbangabe ermöglicht auch die Definition der Deckkraft, wenn eine Füllfarbe oder eine Farbe für die Highlight()- oder Annotate()-Funktion angegeben wird.

Die Deckkraft (auch "Alpha" genannt) ist ein Wert zwischen 0 und 100, wobei 100 bedeutet, dass die Farbe vollständig deckend ist. (Hinweis: Verwechseln Sie dies nicht mit RGB-Werten, die im Bereich von 0...255 liegen).

Die Deckkraft ist optional. Wenn sie nicht angegeben wird, ist sie standardmäßig auf 25 eingestellt. Ansonsten wird sie durch ein Komma von der Farbangabe getrennt.

Beispiele für Farbspezifikationen mit Deckkraft:

"red, 7"

"primary, 75"

"128, 128, 22, 75"

"red, 75~blue, 60"

### Anwendungs-Beispiele:

Highlight("textAbove:Hello Colors!", "blue, 80~SpringGreen, 80");

plotband(high, "green", 2, low, "red", 2, "blue, 40~yellow, 44");

