

Scripting für Indikatoren

Die folgenden Beispiele stehen auch im Scripting-Studio in der Profiversion zur Verfügung und ergeben nach Ausführung folgende Ergebnisse d.h. direkte Logausgabe im Scripting-Studio nach eigener Formatierung und im Watchlisten-Fenster die korrekte automatische Zuordnung und Erstellung der Titel.

The screenshot shows the FormScriptingStudioEdit application interface. The main window is divided into several sections:

- Code Editor:** Contains a script for an RSI indicator. The script includes comments in German, variable declarations, and a procedure named `doSetup` that initializes a watchlist.
- Watchlist (Debug und Testausgaben vom Programm):** A table showing the results of the script execution.
- Code-Beispiele:** A panel on the right showing snippets of code for constructor, procedure, and function declarations.

Code Editor Content:

```
7 am RSI-Indikator, wo eine Signalgenerierung abgefragt wird. Vorhandene
8 Einstellungen werden dabei aus den Chart-Einstellungen übernommen, können
9 aber auch per Parameter-Setup überschrieben werden
10
11 Besonderheiten:
12 - Der Indikator kann allein über die Konstante beim Aufruf CreateIndicatorWithType gewechselt werden
13 - Der Ergebnistyp kann mit CalculateXXXX - Methoden zwischen Ergebniswert, Triggerwert, Aktivierungsgrad etc.
14 jederzeit abgerufen werden
15 - Es können aktiv Watchlisten manipuliert werden, hierzu wird über doSetup eine passende Watchliste, wenn
16 notwendig angelegt und die gefundenen Titel hinzugefügt. Hierbei können dann auch beliebige Kommentare
17 gesetzt werden
18 *)
19
20
21 (* Vorlagerte Variablendefinition ist nur für den Editor notwendig, damit diese erkannt werden und für
22 die Autovervollständigung genutzt werden können *)
23 var
24 itIndicator : TISignalindikator;
25 itStock : TStock;
26 itLogItem : TListItem;
27 itWatchlist : TWatchlist;
28 itWatchItem : TWatchListItem;
29
30 procedure doSetup;
31 begin
32 (* Watchliste anlegen, wenn nicht vorhanden *)
33 itWatchlist := varWatchlists.FindOrCreate('Script-GD-Crossover');
34
35 (* Alle vorhandenen Watchlisten-Einträge entfernen *)
36 itWatchlist.List.Clear;
37
```

Watchlist Content:

Basis	Indicator-Value	Kursanzahl	Signaltyp
2. Gruppe			
<input checked="" type="checkbox"/> Lufthansa vNA F [DE0008232125], Reisen & Freizeit	0,417217046022415	6887	Kaufsignal
<input checked="" type="checkbox"/> Bayer X [DE0008A00017], Chemie	-1	4623	Verkaufssignal

Code-Beispiele Content:

```
begin .. end constructor
begin
end

procedure .. declaration
begin
end

function .. declaration
```

Bezeichnung	Branche	Zeit	Candle	%	Kurs	Stops	CRV	Pos	Kommentar	Vergleich
Bayer`X	Chemie DAX	18.00		0,0 € 0,0%	102,85	Nicht gesetzt	-	0,00	IndSignal:Verkaufsignal	3,3%
Lufthansa vNA`F	Reisen & F DAX	18.00		0,0 € 0,0%	16,17	Nicht gesetzt	-	0,00	IndSignal:Kaufsignal	20,5%

uses Common;

(* CreateDate: 31.07.2014

Author: Jens Werschmoeller

Required: 13.4.18

Ziel ist die Demonstration der Anwendung von Indikatoren. Hier am Beispiel am RSI-Indikator, wo eine Signalgenerierung abgefragt wird. Vorhandene Einstellungen werden dabei aus den Charteinstellungen übernommen, können aber auch per Parameter-Setup überschrieben werden

Besonderheiten:

- Der Indikator kann allein über die Konstante beim Aufruf CreateIndicatorWithType gewechselt werden
- Der Ergebnistyp kann mit CalculateXXXX - Methoden zwischen Ergebniswert, Triggerwert, Aktivierungsgrad etc. jederzeit abgerufen werden
- Es können aktiv Watchlisten manipuliert werden, hierzu wird über doSetup eine passende Watchliste, wenn notwendig angelegt und die gefundenen Titel hinzugefügt. Hierbei können dann auch beliebige Kommentare gesetzt werden

*)

(* Vorlagerte Variablendefinition ist nur für den Editor notwendig, damit diese erkannt werden und für

die Autovervollständigung genutzt werden können *)

var

itIndicator : TISignalindicator;

itStock: TStock;

itLogItem: TListItem;

```

itWatchlist: TWatchlist;
itWatchItem: TWatchListItem;
procedure doSetup;
begin
  { * Watchliste anlegen, wenn nicht vorhanden * }
  itWatchlist:=varWatchlists.FindOrCreate('Script-GD-Crossover');
  { * Alle vorhandenen Watchlisten-Einträge entfernen * }
  itWatchlist.List.Clear;
  (* Header für das Logging passend vorbereiten *)
  varLog.Column[1].Caption:='Indicator-Value';
  varLog.Column[1].Width:=200;
  varLog.Column[2].Caption:='Kursanzahl';
  varLog.Column[2].Width:=200;
  varLog.Column[3].Caption:='Signaltyp';
  varLog.Column[3].Width:=200;
end;
procedure doRun( indicatorTypeID: Integer);
var fSegment: TSegment;
begin
  varLog.BeginUpdate;
  varLog.Items.Clear;
  (* Durchlauf alle Titel zur Prüfung *)
  for idxStock:=1 to varStocks.Count do
  begin
    itStock:=varStocks.Items[idxStock-1];
    (* Verwende nur Titel, die Deutsche Bank im Namen enthalten
      if (Pos('Deutsche Bank',itStock.Name)<>0) then
      if (Pos('DAX',varSegments.GetName( itStock.StockSegments ))<>0) then
      etc.
    *)
    if (itStock.Country='Deutschland') then
    begin
      (* Erzeuge den Indikator hier als den RSI-Indikator *)
      itIndicator:=varIndicators.CreateIndicatorWithTypeID(indicatorTypeID);
      (* Gebe dem Indikator die notwendigen Kursdaten des aktuellen Titels *)
      itIndicator.Kurse:=itStock.Kurs.createCache;
      (* Mittels Indicator.Parameter.Items[1].Wert:=10; könnten die Parameter
        des Indikators verändert werden. Die Bedeutung der Parameter muss pro
        Indikator nachgeschaut werden z.B. über Einstellungen / Indikatorengruppen /

```

Doppelklick auf

```

        den Indikator *)
    (* Berechne die Ergebniswerte für das heutige Datum
        fValue:=itIndicator.CalculateValue(Trunc(Now()), itIndicator.Kurse);
    *)
    fValue:=itIndicator.CalculateErgActivation(Trunc(Now()), itIndicator.Kurse);
    (* Wenn ein Kauf/Verkaufssignal vorliegt, dann Logeintrag + Watch-Eintrag erzeugen *)
    if (fValue<>0) then
    begin
        (* Logging der Ergebnisse *)
        itLogItem:=doLogAndCheck(itStock.Displayname+ ' ['+itStock.ISIN+',
'+itStock.Sector);
        itLogItem.SubItems.Add(FloatToStr(fValue));
        if (itIndicator.Kurse<>nil) then
            itLogItem.SubItems.Add(IntToStr(itIndicator.Kurse.Count)) else
            itLogItem.SubItems.Add('- ');
        if (fValue<0) then
            itLogItem.SubItems.Add('Verkaufssignal') else
            itLogItem.SubItems.Add('Kaufsignal');
        {* Watchlisteneintrag erzeugen *}
        itWatchItem:=TWatchListItem.Create(itStock);
        itWatchItem.CompareDate:=Now;
        if (fValue<0) then
            itWatchItem.Comment:='IndSignal:Verkaufssignal' else
            itWatchItem.Comment:='IndSignal:Kaufsignal';
        itWatchlist.List.Add(itWatchItem);
    end;
    end;
    end;
    varLog.EndUpdate;
end;
begin
    doSetup();
    doRun(cRSI);
end;

```

Revision #1

Created 2022-06-19 07:45:10 UTC by Jens Werschmoeller

Updated 2022-06-19 07:45:41 UTC by Jens Werschmoeller