

Demo-Script

Hier ein sehr einfaches Script, was über alle Aktien iteriert und eine Ausgabe macht, wenn eine RIC[isin] Variable definiert wurde.

Dies ist nur eine Demo-Beispiel, zeigt aber relativ einfach die Funktionsweise. Idee des Scripts war es fehlerhaften Daten-Setups in Aktien zu finden und diese auch automatisch zu korrigieren.

```
uses Common;

(* CreateDate: 10.03.2019
   Author: Jens Werschmoeller
   Required: Shareholder-Version ab 19.3.x (neue Object-Type-Definition notwendig) *)

procedure doSetup;
var lColumn: TListColumn;
begin
    varPanelProgress.Progress.CompletionSmooth:=true;

    (* BeginUpdate/EndUpdate sollte aus Performancegründen zwingend genutzt werden! *)
    varLog.BeginUpdate;
    varLog.Items.Clear;
    (* Header für das Logging passend vorbereiten *)
    varLog.Columns.Clear;
    lColumn:=varLog.Columns.Add; lColumn.Caption:='Name'; lColumn.Width:=350;
    lColumn:=varLog.Columns.Add; lColumn.Caption:='Alter Wert'; lColumn.Width:=200;
    lColumn:=varLog.Columns.Add; lColumn.Caption:='Neuer Wert'; lColumn.Width:=200;

    varLog.GroupView:=false;
    varLog.EndUpdate;
end;

(* Ziel: Automatische Korrektur der RIC-Einträge und gleichzeitig einfachstes Demo-Programm *)
procedure doRun( pIsSimulation: Boolean );
var lStock: TStock;
    lStockVar: TStockVariable;
    lLogItem: TListItem;

begin
```

(* Log-Einträge werden über die varLog-Variable geschrieben, die vom Typ TAdvListView darstellt. Diese greift über .Items auf

alle Log-Zeilen zu. In der Common-Bibliothek sind vereinfachte Methoden-Zugriffe möglich wie doLog *)

```
varPanelProgress.Progress.Max:=varStocks.Count;
```

```
varPanelProgress.Progress.Min:=1;
```

```
varPanelProgress.Progress.ShowGradient:=true;
```

(* Durchlauf alle Titel zur Prüfung *)

```
for idxStock:=1 to varStocks.Count do if (varIsCanceled=false) then
```

```
begin
```

```
  lStock:=varStocks.Items[idxStock-1];
```

(* Zugriff auf den Fortschrittsbalken mit Aktualisierung auf die aktuelle Index-Position *)

```
  varPanelProgress.Progress.Position:=idxStock;
```

(* Aus Performancegründen wird nur bei jedem 500. Durchlauf überhaupt die Anzeige am Frontend aktualisiert *)

```
  if (idxStock mod 500=0) then
```

```
    varApplication.ProcessMessages;
```

(* Zugriff entspricht itStock.StockVariables.ItemsName['RIC[isin]']; Über die ID ist es jedoch "stabiler" *)

```
  lStockVar:=lStock.StockVariables.ItemsNr[12];
```

```
  if (lStockVar<>nil) then
```

```
  begin
```

(* Logging der Ergebnisse über die Hilfsmethode in der Common-Lib / siehe Reiter Gemeinsame Bibliothek *)

```
    itLogItem:=doLogAndCheck(lStock.Displayname+' ['+lStock.ISIN+'], '+lStock.Sector);
```

```
    itLogItem.SubItems.Add(lStockVar.Wert);
```

(* Neue Wertzuweisungen der Variablen sind einfach möglich über itStockWert.Wert = ... *)

```
    itLogItem.SubItems.Add('-');
```

```
  end;
```

```
end;
```

```
end;
```

```
begin
```

```
doSetup;  
doRun( true );  
end;
```

Revision #1

Created 2022-06-19 07:43:04 UTC by Jens Werschmoeller

Updated 2022-06-19 07:43:22 UTC by Jens Werschmoeller