Eigene Programme in MicroStation V8i erstellen

Teil III: Elemente suchen und modifizieren mit VBA

An dieser Stelle wollen wir Ihnen - in einer kleinen Serie - die verschiedenen Möglichkeiten der MicroStation-Programmierung vorstellen. An Hand von kleinen und einfachen Beispielen wollen wir gerade dem "Nichtprogrammierer" oder dem Anfänger einen verständlichen Einblick in die Möglichkeiten der MicroStation Programmierung geben.

Anhand des nachstehenden Beispiels werden wir Ihnen die MicroStation Elementsuche und das Modifizieren der gefundenen Elemente mittels VBA Programm erläutern.

Beispielaufgabe:

Es sollen alle Linien und Linienzüge - die auf der Ebene "Ebene 1" liegen - automatisch verändert werden, dabei sollen die linearen Elemente nach Abschluss des Programms folgende Eigenschaften besitzen:

- Farbe = rot
- Strichstärke = 5
- Strichart = 3

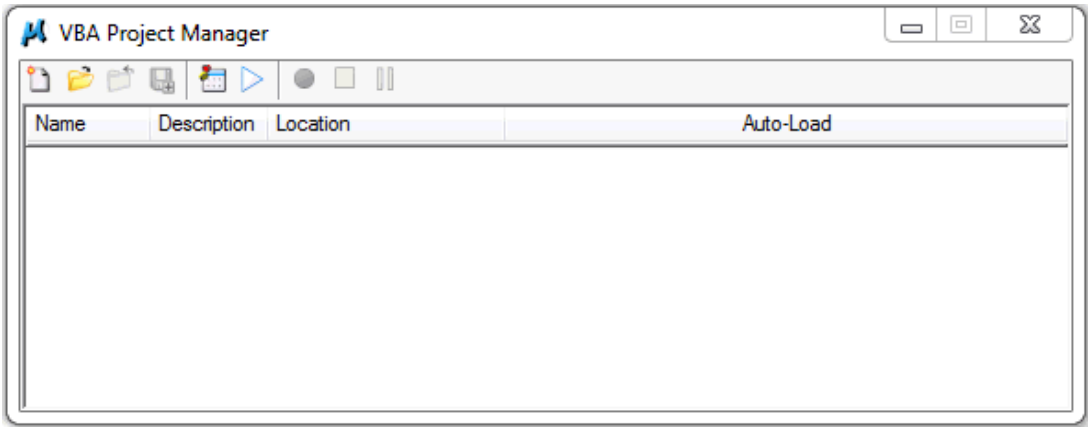
Alle anderen Elemente sollen nicht verändert werden.

Die Hauptaufgaben bestehen in der Einrichtung eines neuen VBA Projekts, dem Einrichten des Elementscanners mit den geforderten Kriterien und in den Modifikationsanweisungen der Elemente.

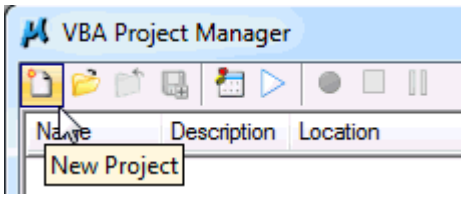
VBA Projekt Manager und IDE:

Über den VBA Projekt Manager wird ein neues VBA Projekt erstellt.

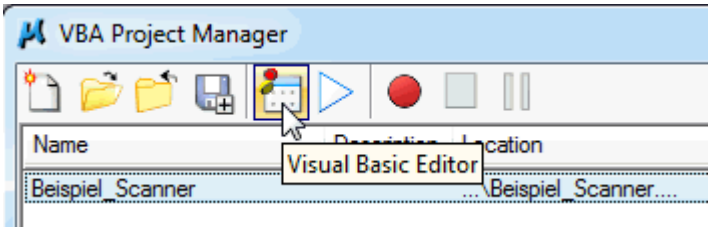
- Öffnen Sie über "Extra/Makro/Projekt Manager" den VBA Projekt Manager



- Über die nachstehende Schaltfläche "Neues Projekt" erzeugen Sie ein neues VBA Projekt

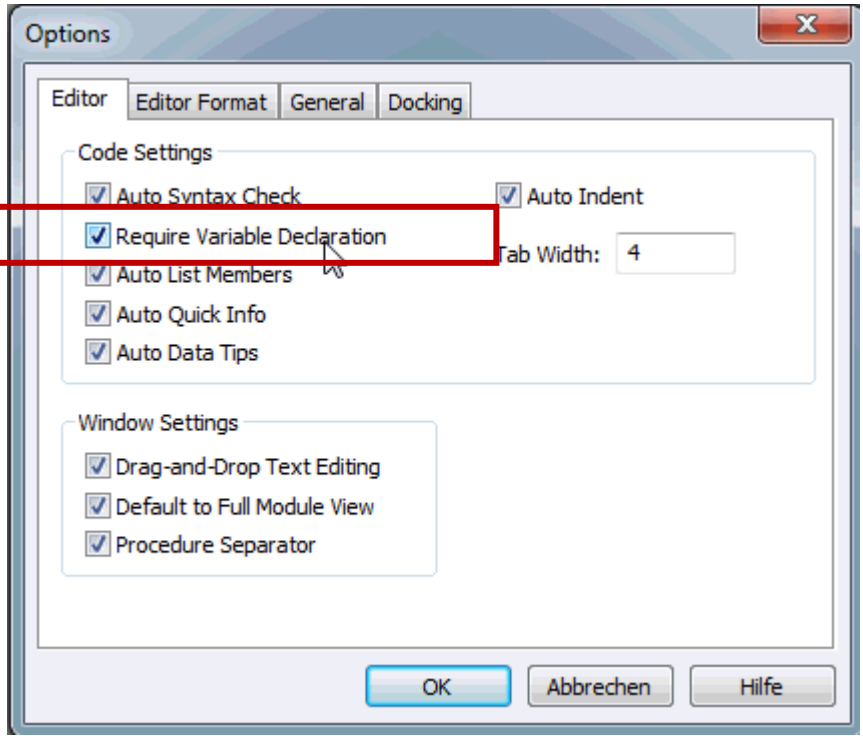


- Speichern Sie das Projekt unter dem Namen "Beispiel_Scanner.mvba" in einem Ordner Ihrer Wahl
- Über die Schaltfläche "Visual Basic Editor" gelangen Sie in die VBA Entwicklungsumgebung IDE (Visual Basic for Applications Integrated Development Environment)



Vor dem eigentlichen Start sollte eine wichtige Einstellung vorgenommen werden, die dafür sorgt, dass Regeln bezüglich der Variablenverwendung bei der Erstellung des Codes eingehalten werden.

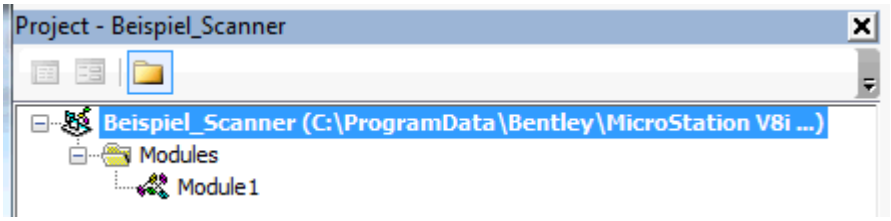
- Über das Menü "Tools/Options" gelangen Sie in den nachstehenden Dialog, in dem Sie die Option "Require Variable Declaration" bitte anhaken



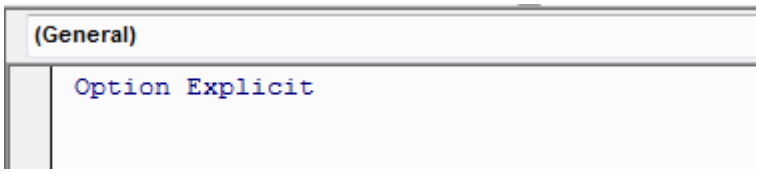
- Die Option "Require Variable Declaration" besagt, dass Sie vor der Verwendung einer Variablen diese zunächst definieren müssen.

VBA Programm:

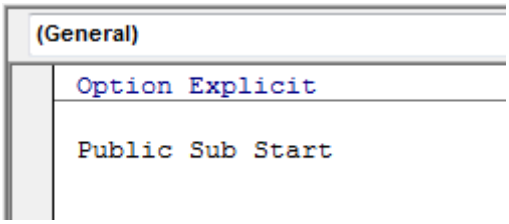
Jetzt sind die wichtigsten Einstellungen gemacht und es kann endlich losgehen. Im linken Teil der IDE (im Projektexplorer) befindet sich das eben neu angelegte Projekt mit dem Namen "Beispiel_Scanner". Durch einen Doppelklick auf das Projekt bzw. den Projektnamen öffnet sich der "Projektbaum", in dem sich standardmäßig ein Modul mit dem Namen "Module1" befindet.



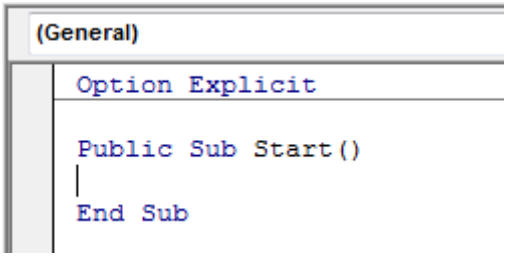
Auf der rechten Seite der IDE befindet sich das Text- bzw. Programmcode Fenster.



- Tragen Sie hier bitte den nachstehenden Code ("Public Sub Start") ein und bestätigen Sie Ihre Eingabe mit einem <Enter>



- Die VBA Engine vervollständigt automatisch Ihre Eingabe, siehe das Klammerzeichen "()" und das Ende der Funktion "End Sub".



- Übertragen Sie bitte den nachstehenden Code
- Allgemeine Anmerkungen zur besseren Lesbarkeit des Quellcodes werden durch ein Hochkomma eingeleitet

Programmcode:

Zunächst müssen einige Variablen definiert werden

- Die Variablen "ele", "lv" und "ls" repräsentieren das MicroStation Element, die Ebene und die Strichart
- Die Variable "oScan" stellt die Suchmaske mit den später noch zu definierenden Elementkriterien dar
- Die Variable "ee" repräsentiert eine Elementaufzählungsliste der gefundenen Elemente ("Trefferliste")

```
Public Sub START()

    Dim ee As ElementEnumerator
    Dim oScan As ElementScanCriteria
    Dim lv As Level
    Dim ele As Element
    Dim ls As LineStyle
```

An dieser Stelle wird eine neue Suchmaske definiert. Mit Hilfe der Suchmaske werden später die relevanten MicroStation Elemente gesucht und gefunden.

- Mit den Befehlen "oScan.ExcludeAllLevels" und " oScan.ExcludeAllTypes" werden alle Ebenen und Elementtypen aus der Suchmaske entfernt

```
    ' Neue Suchmaske definieren
    Set oScan = New ElementScanCriteria

    ' Zuerst alle Ebenen aus der Suchmaske entfernen
    oScan.ExcludeAllLevels

    ' Alle Elementtypen aus der Suchmaske entfernen
    oScan.ExcludeAllTypes
```

In der Aufgabenbeschreibung hieß es "Suche .... auf der Ebene "Ebene 1"".

An dieser Stelle wird sichergestellt, dass in der Ebenendefinition der DGN Zeichnung tatsächlich eine "Ebene 1" existiert. Ist in der Zeichnung keine "Ebene 1" definiert, wird das Programm an dieser Stelle abgebrochen.

```
    ' Ebene 1 im aktuellen Designfile suchen
    Set lv = ActiveDesignFile.Levels.Find("Ebene 1")

    If lv Is Nothing Then
        ' Wurde die Ebene 1 nicht gefunden, Programmabbruch
        Exit Sub
    End If
```

Anschließend muss die Suchmaske mit entsprechenden Kriterien versehen werden.

- Das erste Kriterium ist die Einschränkung der Suche auf die "Ebene 1"
- Ein weiteres Kriterium ist die Einschränkung der Suchmaske auf die Elementtypen Linie und Linienzug

- Nach der Definition der Kriterien muss die eigentliche Elementsuche (das "Scannen") im aktiven Modell ausgeführt werden
- Das Ergebnis der Suche ist eine Liste von Elementen ("Trefferliste"), die anschließend Element für Element abgearbeitet werden muss

```
    ' Ebene 1 als Kriterium der Suchmaske setzen
    oScan.IncludeLevel lv

    ' Linien als Kriterium der Suchmaske setzen
    oScan.IncludeType msdElementTypeLine

    ' Linienzüge als Kriterium der Suchmaske setzen
    oScan.IncludeType msdElementTypeLineString

    ' Aktives Model scannen, Liste von Ergebniselementen
    Set ee = ActiveModelReference.GraphicalElementCache.Scan(oScan)
```

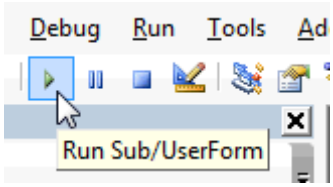
Über eine Schleife ("Do While / Loop"), die solange durchlaufen wird, bis es in der Trefferliste keine weiteren Elemente mehr gibt, wird jedes Element, welches den Kriterien der Suchmaske entspricht, bearbeitet.

- Setzen der Elementfarbe auf 3
- Setzen der Linienstärke auf 5
- Beim Setzen der Linienart ist eine Sicherheitsabfrage notwendig, die sicherstellt, dass es die zu setzende Linienart tatsächlich gibt
- Zu guter Letzt wird das modifizierte MicroStation Element gespeichert und in die DGN Zeichnung zurückgeschrieben

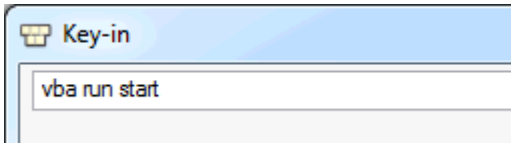
```
    ' Gefundene Elemente durchlaufen
    Do While ee.MoveNext
        Set ele = ee.Current

        ' Setze Elementfarbe auf 3
        ele.Color = 3

        ' Setze Elementstrichstärke auf 5
        ele.LineWeight = 5

        ' Setze Elementlinienart auf 3
        Set ls = ActiveDesignFile.LineStyles.Find("3")
        If Not (ls Is Nothing) Then
            Set ele.LineStyle = ls
        End If

        ' Geändertes Element speichern
        ele.Rewrite
    Loop
End Sub
```

Programmaufruf:

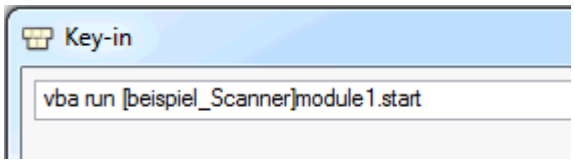
- Über den Menüeintrag "Debug/Compile Beispiel_Scanner" wird der Quellcode formal auf Richtigkeit geprüft.
- Sollte ein Fehler im Programmcode vorliegen, wird dieser über einen Dialog angezeigt und muss bereinigt werden
- Über die Schaltfläche "Run Sub/UserForm" kann das Programm direkt ausgeführt werden



- alternativ kann das MVBA Programm über das MicroStation KeyIn oder Eingabefenster aufgerufen und gestartet werden
- befindet sich das MVBA in dem MicroStation Suchpfad und verwenden Sie nicht mehrere VBAs mit einer Funktion "Start", reicht der nachstehende Programmaufruf in der Regel aus



- sollten Sie mehrere MVBAs mit der Funktion "Start" verwenden, sollten Sie die Eingabe - wie nachstehend aufgeführt- verfeinern



- Haben Sie alles richtig gemacht, wird die DGN Zeichnung automatisch durchsucht und alle Linien und Linienzüge, die auf der Ebene "Ebene 1" liegen, werden dick, rot und gestrichelt erscheinen.