

Generierung und Auswertung von Wortlisten aus VB-Programmen. Für die Ausführung muss zuvor «ProjectSummary1» gelaufen sein¹.

Inhalt

1. Vorbemerkung	2
2. Wortliste anzeigen	2
3. Wortliste generieren	3
4. Projekte	4
5. Dateien	5
6. NotePad++	6

¹ Für die Speicherorte massgebend sind die Angaben ex «ProjectSummary1.txt» im Programmordner von ProjectSummary1.

1. Vorbemerkung

Die Anwendung generiert eine Auflistung von Worten, welche aus den Programmen extrahiert werden. Im Programmordner dieser Anwendung ist eine Datei «Filter.txt» gespeichert. Dort gespeicherte Worte werden bei der Generierung ausgefiltert².

Mit [F1] wird das PDF-Dokument für die Hilfe aufgerufen.

Auf allen Fenstern mit einer Auflistung wird nach Eingabe eines Buchstabens die erste Zeile selektiert, bei welcher der Anfangsbuchstabe übereinstimmt (gross/klein ohne Belang).

Wird ein Fenster minimiert, maximiert oder geschlossen, erfolgt dasselbe bei allen untergeordneten Fenstern.

2. Wortliste anzeigen

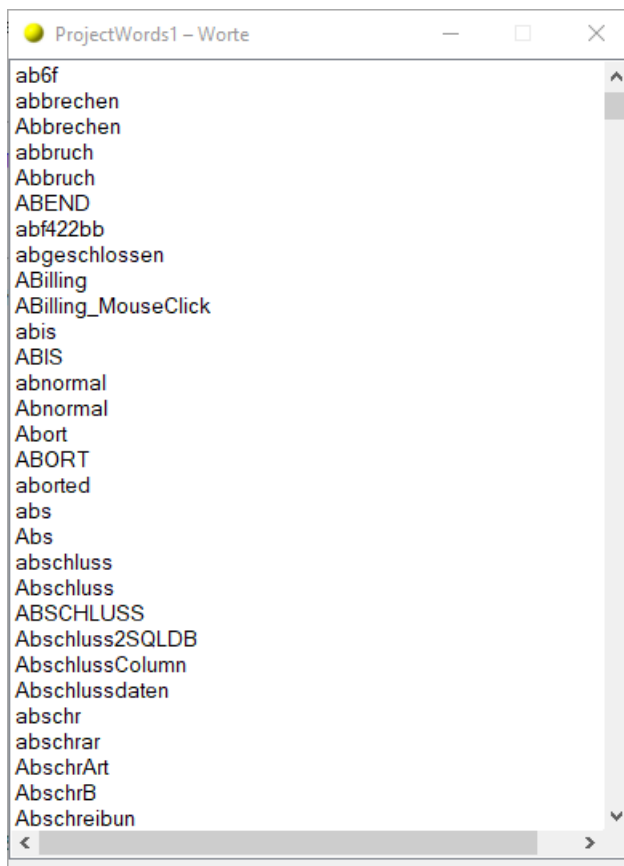


Abbildung 1: Worte

[Ctrl] öffnet das Fenster zu Suche. Mit [Enter] bzw. mit Schliessen des Fensters, startet die Suche, welche auf Übereinstimmung in den Anfangspositionen ausgerichtet ist.

Falls beim Start noch keine Wortliste existiert, wird die Generierung automatisch gestartet (Seite 3).

Andernfalls kann der Benutzer entscheiden.

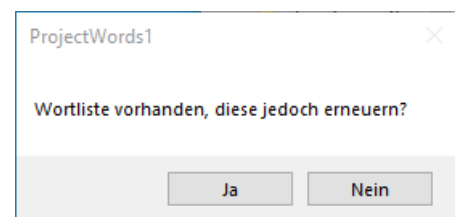


Abbildung 2: Benutzeraktion

Eine Erneuerung ist sinnvoll, wenn zwischenzeitlich Programme geändert haben.

Mausklick in eine Zeile zeigt die Projekte an, in welchem das Wort vorkommt (Seite 4).

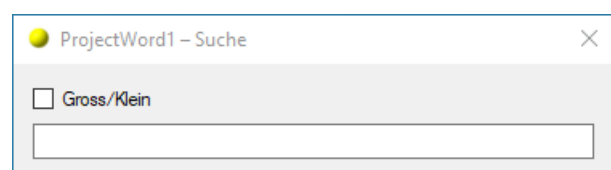


Abbildung 3: Suche

² Gross-/Kleinschrift von Belang.

3. Wortliste generieren

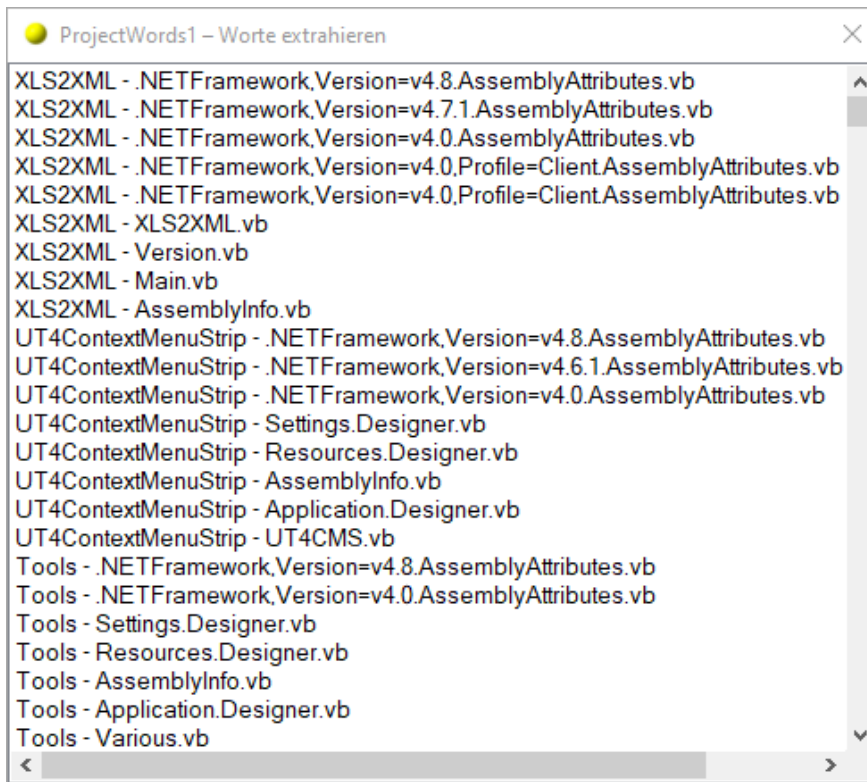


Abbildung 4: Worte extrahieren

Ausgewertete Dateien³ werden laufend und absteigend gelistet. Als Ergebnis ist die Datei «WordList.txt» im Ausgabeordner von «ProjectSummary1» abgelegt.

Vom «Windows Form Designer» generierter Code sowie Kommentar in den Dateien werden nicht berücksichtigt.

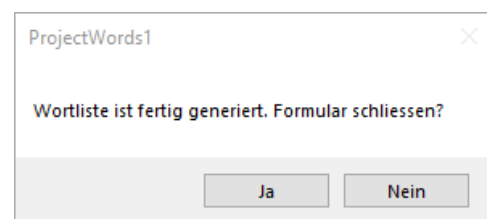
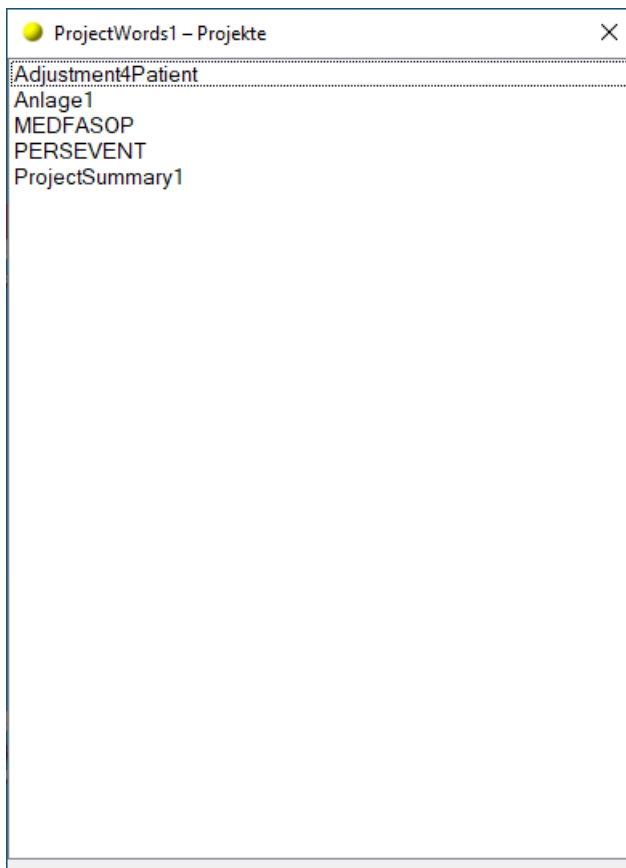


Abbildung 5: Benutzeraktion

³ Datei «ProjectSummary1.txt» im Temp-Ordner bestimmt die auszuwertenden Projekte.



4. Projekte



Mausklick in eine Zeile zeigt die Dateien an, in welchem das Wort vorkommt (Seite 5).

Abbildung 6: Projekte

5. Dateien



Mausklick in eine Zeile ruft den Editor «Notepad++»⁴ auf, mit der Datei im Lese-modus.

Das gesuchte Wort ist im Zwischenspeicher und kann von dort mit [CtrlV] abgerufen werden.

Abbildung 7: Dateien

⁴ NotePad++ ist zwingend notwendig.



6. NotePad++

```
1 Public Class Form1
2     Private vbtabs As ProjectTab
3     Public t4f As Tool4Form.Tool4Form
4     Public t4p As Tool4Param.Tool4Param
5     Private enc As System.Text.Encoding
6     Private sw As IO.StreamWriter
7     Private swprf As IO.StreamWriter
8     Private swdll As IO.StreamWriter
9     Public inp As String = ""
10    Public out As String = ""
11    Private doc As String
12    Private dlltab As BTree1.Table
13    Private sdll As String
14    Private sqtxt As String = ""
15    Private uddl As New Usage4DLL
16    Public vsprog As String
17    Private Class cdll
18        Inherits BTree1.Data
19        Public name As String
20        Public Overrides Function Compare(ByRef d As BTree1.Data) As Short
21            Compare = String.Compare(name, CType(d, cdll).name)
22        End Function
23        Public Overrides Sub Apply(Optional ByRef sdll As Object = Nothing)
24            If CType(sdll, String).Length > 0 Then sdll &= ", "
25            sdll &= name
26        End Sub
27    End Class
28
29    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles Me.Load
30        Dim s As String
31        CheckLicense()
32        t4f = New Tool4Form.Tool4Form(Application.StartupPath, Application.ProductName)
33        t4f.Text2Form(Me, ToolTip1)
34        t4p = New Tool4Param.Tool4Param(Application.StartupPath, Application.ProductName)
35        inp = t4p.GetParam("input")
```

Abbildung 8: NotePad++

Das gesuchte Wort ist im Zwischenspeicher und kann von dort mit [CtrlV] abgerufen werden.