

PB TermLister

Externes Tool für PureBasic 5.20

Charly Bauer

Basiert auf dem veröffentlichten Quellcode von Bernd (BSP) vom PureBasic-Forum Germany!

Quick Guide V0.20 (Windows & Linux)

Ergänzungen/Änderungen gegenüber dem veröffentlichten Quellcode ...

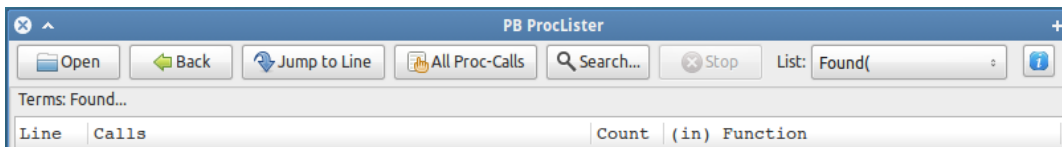
- Programm aufgehübscht, u.a. mit PureBasic-Icons ;-)
- Änderung der Listendarstellung von EditorGadget in ListIconGadget mit Tabellendarstellung.
- Änderung des Menüs in eine ComboBox.
- Verbesserung (immer noch lückenhaft) der Prozedur- und Prozeduraufruferkennung.
- Aufnahme der Suche weiterer Quellcodeausdrücke (Terms) wie Variablen, Konstanten, Zeiger ...
- Kennzeichnung unterschiedlicher Schreibweisen (Groß-/Kleinschrift).
- Kennzeichnung fehlender Deklarationen und Anlage einer passenden Deklarationsliste im Clipboard.
- Änderung einiger Windows-API-Aufrufe in PureBasic-eigene Möglichkeiten.
- Zusätzliche Auswertung von %WORD + %SELECTION-Parametern um problematische Erkennungen zu verbessern.

Features ...

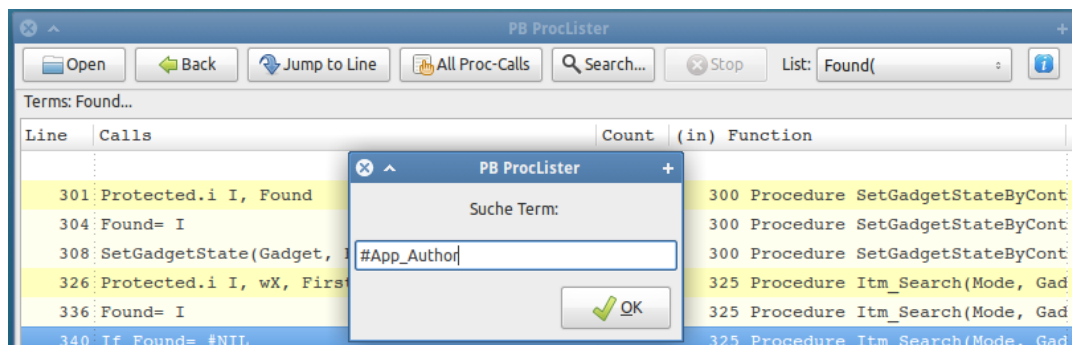
- Arbeitet als externes Werkzeug / Tool für die Programmiersprache PureBasic - z.Zt. unter Windows und Linux. Die Mac-Version kann mangels Mac nicht angepasst werden!
- Listet alle **Prozeduren** aller **Includes** anhand der ¹Hauptdatei eines wählbaren und übergebenen Projektes oder des aktiven Quellcodes.
- Listet alle **Aufrufe** einer daraus gewählten oder im Editor per Block bzw. Cursorposition gekennzeichneten Prozedur oder eines Ausdruckes.
- Markiert nötige **Deklarationen** von Prozeduren.
- Markiert alle nicht aufgerufenen (**überflüssigen**) Prozeduren.
- Listet **alle Aufrufe aller Prozeduren** eines Projektes.
- **Exportiert** alle nötigen **Deklarationen** aus der vorher genannten Kompletliste in das Clipboard.
- Listet **Term-Aufrufe (Variablen)**, die per *Suchen* eingegeben oder per Editor-Block bzw. Cursorposition übergeben wurden, sofern dieser als Prozedur nicht gefunden wurde.
- Kennzeichnet die **Deklaration der Terme oder Prozeduren**.
- **Springt** an jede gelistete Stelle im PureBasic-Editor (unter Linux noch problematisch).

¹ Hauptdatei= Sourcecodedatei bzw. Hauptdatei in Projekten, d.h. die Datei unter Eingabe-Quellcode in den Projekt-Compiler-Optionen

1. Das Menu (Toolbar)



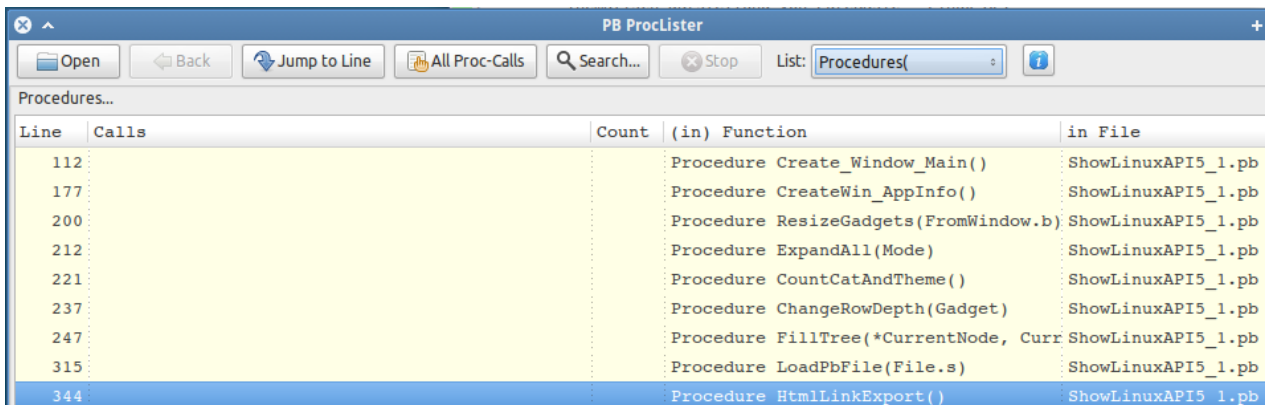
- [Open]** öffnet per Fileselektor eine neue Hauptdatei eines Projektes.
Alternativ bzw. eigentlich wird diese Datei durch die aktive Quellcodeseite beim Aufruf aus PureBasic übergeben.
- [Back]** schließt die aktuell gezeigte Liste und zeigt die Vorherige, siehe auch Combobox 'List'.
- [Jump to Line]** springt anhand der selektierten Listenzeile zu PureBasic in die Datei und die Zeile des Quellcodes.
Dies startet unter Linux noch eine neue PB-Instanz und ist noch nicht gelöst!
- [All Proc-Calls]** listet in der aktuellen Hauptdatei alle Prozeduren mit allen Aufrufen.
- [Search]** öffnet einen Eingabe-Dialog und listete alle Aufrufe:
zuerst als möglicher Prozeduraufruf. Falls ohne Fundstelle ...
dann als Variable.



- [List]** Die Combobox listet der Reihe nach alle offenen Listen. Per Auswahl wird diese direkt angezeigt.

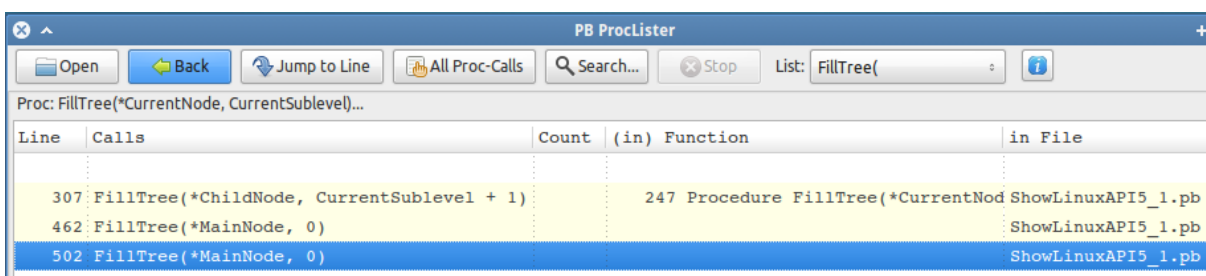
2. Haupt-Prozedurliste

Nach dem Start erscheint zuerst eine Auflistung der Prozeduren im Quellcode ...



3. Folge

3.1 Prozeduraufrufe



3.2 Variablenverwendung (Terms)

Line	Calls	Count	(in) Function	in File
40	#Main_NodeToggle			ShowLinuxAPI5_1.pb
116	ButtonImageGadget(#Main_NodeToggle, 10, 4, 26,		112 Procedure Create_Window_Main()	ShowLinuxAPI5_1.pb
118	SetGadgetAttribute(#Main_NodeToggle, #PB_Button		112 Procedure Create_Window_Main()	ShowLinuxAPI5_1.pb
121	GadgetToolTip(#Main_NodeToggle, "Opens and clos		112 Procedure Create_Window_Main()	ShowLinuxAPI5_1.pb
167	SetGadgetState(#Main_NodeToggle, #True)		112 Procedure Create_Window_Main()	ShowLinuxAPI5_1.pb
489	Case #Main_NodeToggle			ShowLinuxAPI5_1.pb
490	If GetGadgetState(#Main_NodeToggle)			ShowLinuxAPI5_1.pb
504	SetGadgetState(#Main_NodeToggle, #True)			ShowLinuxAPI5_1.pb

3.3 Folgelisten

... können durch Doppelklick auf Zeilen mit Prozedureinträgen in der '(in) Funktion'-Spalte beliebig weiter mit der Auflistung dieser Prozeduraufrufe geöffnet werden.

3.4 Kennzeichnungen

- Die Zeile der Deklaration wird in kräftigerer Farbe gekennzeichnet.

Line	Calls	Count	(in) Function
87	Global.i I, gEvent, gEventWin, gQuit, gNumb, gC		
188	TextGadget(#INF_T5, 5, 127, 260, 18, Str(gNumb)		177 Procedur
231	gNumb+ 1		221 Procedur

- Aufrufe in roter Schrift (Spalte 'Calls') unterscheiden sich gegenüber dem vorgegebenen Ausdruck (Groß-/Kleinschrift)

2439	Procedure.1 xlsFunc_GetLastRow(ExcelObject....	3	
1554	ExcLastRow= xlsFunc_GetLastRow(myExcObj)		153
1594	ExcLastRow= xlsFunc_GetLastRow(myExcObj)		153
355	Declare.1 XLSFunc_GetLastRow(ExcelObject.CO...		

4. Aufrufe aller Prozeduren listen

```

3808 Procedure.d Min(A.d, B.d) 4
1395 ViewPos= Min(Run\Counter+ (VisItems* 0.6), Items) 1336 Procedure RunProgramm() COM-Tool48.pb
1517 For I= 0 To Min(gVisiblePorts, gComPorts- 1) 1515 Procedure CheckPortStates(All... COM-Tool48.pb
2164 For O= 1 To Min(POutTable\Columns, POutTable\MaxColumns) 2145 Procedure TableClear (Type) COM-Tool48.pb
579 Declare.d Min(A.d, B.d) COM-Tool_Header.pb

3814 Procedure.d Max(A.d, B.d) 6
2036 POutTable\Columns = Max(Val(GetGadgetText(#OCF_PkCols)), 1) 2023 Procedure.i ProtSettingsAssume() COM-Tool48.pb
2037 POutTable\Width = Max(Val(GetGadgetText(#OCF_PkColWidth)), 10) 2023 Procedure.i ProtSettingsAssume() COM-Tool48.pb
2038 POutTable\MaxColumns= Max(Val(GetGadgetText(#OCF_PkMaxCols)), 1) 2023 Procedure.i ProtSettingsAssume() COM-Tool48.pb
2039 POutTable\MaxRows = Max(Val(GetGadgetText(#OCF_PkMaxRows)), 1) 2023 Procedure.i ProtSettingsAssume() COM-Tool48.pb
2223 POutTable\Columns= Max(ActCols, POutTable\Columns) 2220 Procedure TableAddColumn() COM-Tool48.pb
578 Declare.d Max(A.d, B.d) COM-Tool_Header.pb

```

Hier werden alle Prozeduren und deren Verwendung aufgelistet.

4.1 Kennzeichnungen

- Durch einen roten Hintergrund der Zeilennummer von Prozedur-Zeilen werden **zu deklarierende** Prozeduren gekennzeichnet, siehe obige Tabelle.
- Durch einen hellroten Hintergrund ganzer Prozedur-Zeilen werden ungenutzte Prozeduren gekennzeichnet.

Line	Calls	Count	(in) Function
146	Procedure TestLeer()	0	
153	Procedure WindowOpen()	1	
1017	WindowOpen()		

- Aufrufe in roter Schrift unterscheiden sich gegenüber dem vorgegebenen Term (Groß-/Kleinschrift)

```

2439 Procedure.i xlsFunc_GetLastRow(ExcelObject.... 3
1554 ExcLastRow= xlsFunc_GetLastRow(myExcObj) 153
1594 ExcLastRow= xlsFunc_GetLastRow(myExcObj) 153
355 Declare.i XLSFunc_GetLastRow(ExcelObject.CO...

```

4.2 Deklarationsliste

Im Clipboard wird nach Erstellung dieser Kompletliste eine nach Includes bzw. Files sortierte Liste aller zu deklarierenden Prozeduren angelegt. Diese kann per **[Strg][V]** direkt ins Programm oder einen Texteditor zum Vergleich übernommen werden.

5. Aufruf aus PureBasic

- In PureBasic das Tool in Ihr Verzeichnis für externe Werkzeuge kopieren.
- Unter **Werkzeuge / Werkzeuge konfigurieren** mit **[Neu]** den Dialog **Werkzeugeinstellungen bearbeiten** aufrufen.



- Unter **>Werkzeugeinstellungen bearbeiten<** den Dialog entsprechend Ihrer Installation und dem gewünschten Tastaturkürzel nach folgendem Beispiel ausfüllen, bei 64-Bit-Systemen die Datei **PB TermListerx64**, sonst **PB TermLister** in der Kommandozeile angeben ...

5.1 Liste eines Ausdrucks (Proc/Term) nach Cursorposition aufrufen

Im folgenden Beispiel nach der Variablen 'Proc'.

```

Procedure i Is_Proc(Line.s, Proc.s)
  Protected i Okay= #Null, Sgn= 1
  Okay= FindString(Line, Proc, 1, #PB_String_NoCase)
  If Okay
    If Not FindString(Line, Proc) : Sgn= -1 : EndIf
  EndIf
  If Okay > 1
    If FindString(#ProcPre, Mid(Line, Okay-1, 1))
      Okay= 1
    Else
      Okay= 0
    EndIf
  EndIf
  ProcedureReturn Okay* Sgn
EndProcedure

```

Diese Form des Aufrufes ist einfacher, liefert jedoch z.B. bei Konstanten die führende Raute nicht mit, wodurch der Ausdruck nicht oder falsch erkannt wird. Ein PureBasic-Problem mit %WORD-Parameter

5.2 Liste eines Ausdrucks (Proc/Term) nach Block aufrufen

```

Procedure i Is_Proc(Line.s, Proc.s)
  Protected i Okay= #Null, Sgn= 1
  Okay= FindString(Line, Proc, 1, #PB_String_NoCase)
  If Okay
    If Not FindString(Line, Proc) : Sgn= -1 : EndIf
  EndIf
  If Okay > 1
    If FindString(#ProcPre, Mid(Line, Okay-1, 1))
      Okay= 1
    Else
      Okay= 0
    EndIf
  EndIf
  ProcedureReturn Okay* Sgn
EndProcedure

```

Diese Form des Aufrufes ist selektiver und kann auch Konstanten korrekt übergeben, funktioniert jedoch nicht, wenn der Block an das Zeilenende grenzt, da PureBasic hier einen ungültigen Blockbereich bei **%SELECTION**-Parameter liefert. Ansonsten wird von PureBasic das Blockende auch als Cursorposition übernommen und der Ausdruck übergeben. (Das Problem betrifft V5.22 und kann zwischenzeitlich behoben worden sein!)

6. Geht (noch) nicht - wäre jedoch wünschenswert

- Windows-Version ist z.Zt. weniger entbuggt.
- Arbeitet bisher nur mit Ascii-Code (denke ich?).
- Zusätzliche Lauffähigkeit auf Mac
- Der Rücksprung in die Editorzeile öffnet auf Linux stets das PureBasic neu. Der PB-Editor-Schalter 'nur eine Instanz starten' ist auf Linux wirkungslos und vermtl. generell noch ein Problem. Einem 'fremden' Scintilla einen Befehl zukommen zu lassen ist mir ebenfalls noch nicht gelungen.
- ... und Quellcode wieder plattform-unabhängig in einem Quelltext gestalten.
- Selbstständige Erkennung und Auflistung von Konstanten
- Selbstständige Erkennung und Auflistung von Makros
- Selbstständige Erkennung und Auflistung von Variablen (Global oder Lokal)
- Selbstständige Erkennung und Auflistung von Strukturen und deren Feldern
- Selbstständige Erkennung und Auflistung von LinkedLists
- Selbstständige Erkennung und Auflistung von Maps
- Selbstständige Erkennung und Auflistung von Arrays
- Selbstständige Erkennung und Auflistung von Modulen
-
- Änderungen am Quellcode:
 - Sichere Umbenennung von Variablen (kann einem per PureBasic Suchen/Ersetzen-Dialog nach ziemliche Probleme bereiten)
 - Groß-/Kleinschreibungsabgleich von Ausdrücken gegenüber der Deklaration
 - Bereinigung von Zeilenenden