

PBMakeTool

Einführung:

PBMakeTool arbeitet ähnlich wie das Programm make aus Linux, allerdings speziell ausgelegt für Tools und Precompiler, die für PureBasic geschrieben wurden. Anhand einer „PBMake.txt“ startet PBMakeTool die angegebenen Programme bei Ereignissen, die in der PB-IDE stattfinden, wie z.b. das Kompilieren einer Datei.

Der Vorteil an PBMakeTool besteht darin, dass die Programme nur dann ausgeführt werden, wenn die Output-Datei älter ist als die Ausgangsdatei, bzw. gar nicht vorhanden ist. Vor allem bei den Generatoren PBParserGen und PBLexerGen macht dies durchaus Sinn, da ein Programm in der Regel weiter ausgebaut wird, wenn der Parser und Lexer bereits fertig sind. So müssen diese nicht bei jedem Kompilieren erneut generiert werden.

Arbeitsweise:

PBMakeTool prüft bei jedem Kompilieren einer Datei aus der PB-IDE, ob es im gleichen Ordner eine Datei mit dem Namen „PBMake.txt“ gibt. Wenn dies der Fall ist, befolgt es die Anweisungen, die dort angegeben wurden.

Die Syntax einer solchen PBMake.txt ist wie folgt:

```
Trigger1 [, Trigger2, ...]:  
    Befehl1 Parameter1 Parameter2 ...  
    Befehl2 ...  
    ...  
;  
Trigger3 [, Trigger4, ...]:  
    ...
```

Wobei Trigger eines der folgenden sein kann:

- BEFORE_EXE: Vor dem Erstellen des Executable
- BEFORE_COMP: Vor dem Kompilieren/Starten
- AFTER_EXE: Nach dem Erstellen des Executable
- AFTER_COMP: Nach dem Kompilieren/Starten

Ein Beispiel, wie eine solche Datei aussehen könnte:

```
BEFORE_EXE, BEFORE_COMP:  
    PBParserGen parser.pb -o parser_out.pb  
;
```

Beispiel

Falls in der PB-IDE eine Datei kompiliert oder ein Executable erstellt wird, ruft PBMakeTool das Programm PBParserGen auf, mit den Parametern „parser.pb“, „-o“ und „parser_out.pb“, aber auch nur dann, wenn die Datei „parser_out.pb“ älter ist, als „parser.pb“

Automatische Installation:

- Das Programm PBMakeToolInstaller.exe starten
- Hinter dem Text „Tools.prefs“ die gleichnamige Preference-Datei der PB-IDE wählen, falls nicht ohnehin schon der richtige Pfad angegeben wurde
- Auf „Installiere PBMakeTool“ klicken

Manuelle Installation:

Für die manuelle Installation 4 neue Werkzeuge erstellen:

1.
Name: PBMakeTool BEFORE_EXE
Argumente: BEFORE_EXE „%PATH“
Arbeitsverzeichnis:
Ereignis zum Ausl.: Vor dem Erstellen des Executable
Flags: „Warten bis zum Beenden“, „Werkzeug [...] verstecken“
2.
Name: PBMakeTool BEFORE_COMP
Argumente: BEFORE_COMP „%PATH“
Arbeitsverzeichnis:
Ereignis zum Ausl.: Vor dem Kompilieren/Starten
Flags: „Warten bis zum Beenden“, „Werkzeug [...] verstecken“
3.
Name: PBMakeTool AFTER_EXE
Argumente: AFTER_EXE „%PATH“
Arbeitsverzeichnis:
Ereignis zum Ausl.: Nach dem Erstellen des Executable
Flags: „Warten bis zum Beenden“, „Werkzeug [...] verstecken“
4.
Name: PBMakeTool AFTER_COMP
Argumente: AFTER_COMP „%PATH“
Arbeitsverzeichnis:
Ereignis zum Ausl.: Nach dem Kompilieren/Starten
Flags: „Warten bis zum Beenden“, „Werkzeug [...] verstecken“

Wobei die beiden Werkzeuge 3 und 4 von den Generatoren nicht gebraucht werden. Sie sind erst mal nur zur Vollständigkeit halber vorhanden. Sie sind also z.Z. Nicht zwingend notwendig.