

Dialog Maker (unfertig)

Eine kleine Anleitung.

27. Aug. 2009

Inhaltsverzeichnis

Dialog Maker (unfertig).....	1
Beschreibung.....	1
Hinweise.....	1
Pool bearbeiten.....	3
Legende.....	3
Design Editor.....	5
Legende.....	5
Menü.....	7
Menü Gadgets.....	7

Beschreibung

Dialog Maker ist ein einfaches Programm, um bei der Erstellung von Dialogen in PureBasic zu helfen (und evtl. auch anderen Sprachen). Hauptaugenmerk war die Flexibilität bezüglich der Dialogelement-Typen - diese können mitsamt generiertem Code frei selber erstellt werden!

Die Dialogelemente werden deshalb auch nicht wirklich angezeigt, sondern nur als Boxen mit den entsprechenden Dimensionen dargestellt. Man kann Dialog Maker wohl eher als Programmierwerkzeug sehen, wie als visuelles Tool - man kann damit Codeschnipsel aneinander reihen und diese Schnipsel haben einfach auch noch eine gezeichnete Entsprechung der X-, Y-Koordinate und der Höhe und Breite.

Dafür ist es möglich, jede zukünftige Änderung von Dialogelementen anzupassen oder neue hinzu zu fügen. Unter Umständen wäre es sogar möglich, Dialog Maker für andere Programmiersprachen zu verwenden (wobei dies bei der Entwicklung nicht besonders beachtet wurde).

Dialog Maker erzeugt aus den zusammen gefügten Codeschnipseln eine Datei, welche in einen Hauptquelltext per "Include" eingefügt werden kann. Die erzeugte Datei kann nicht bearbeitet werden - sie wird immer wieder überschrieben.

Bei der Entwicklung wurde versucht, möglichst keine API-Aufrufe zu verwenden, damit der Code leichter portierbar bleibt. Deshalb ist das Programm zum Teil auch etwas speziell zu bedienen.

Hinweise

Für Fehler oder Schäden, die durch eventuelle Programmfehler auftreten könnten, übernimmt der Autor keine Haftung. Das Programm ist kostenlose Freeware. Die kommerzielle Verwertung des Programms (Verkauf, Vertrieb der Software etc...) ist nicht gestattet.

Die Weitergabe im nicht kommerziellen, privaten und schulischen Rahmen ist gestattet und

erwünscht.

Die Weitergabe auf kommerziellen Wegen ist nur mit Genehmigung des Autors gestattet.

Andy Marschner

Schweiz

<http://www.marschner.ch/software>

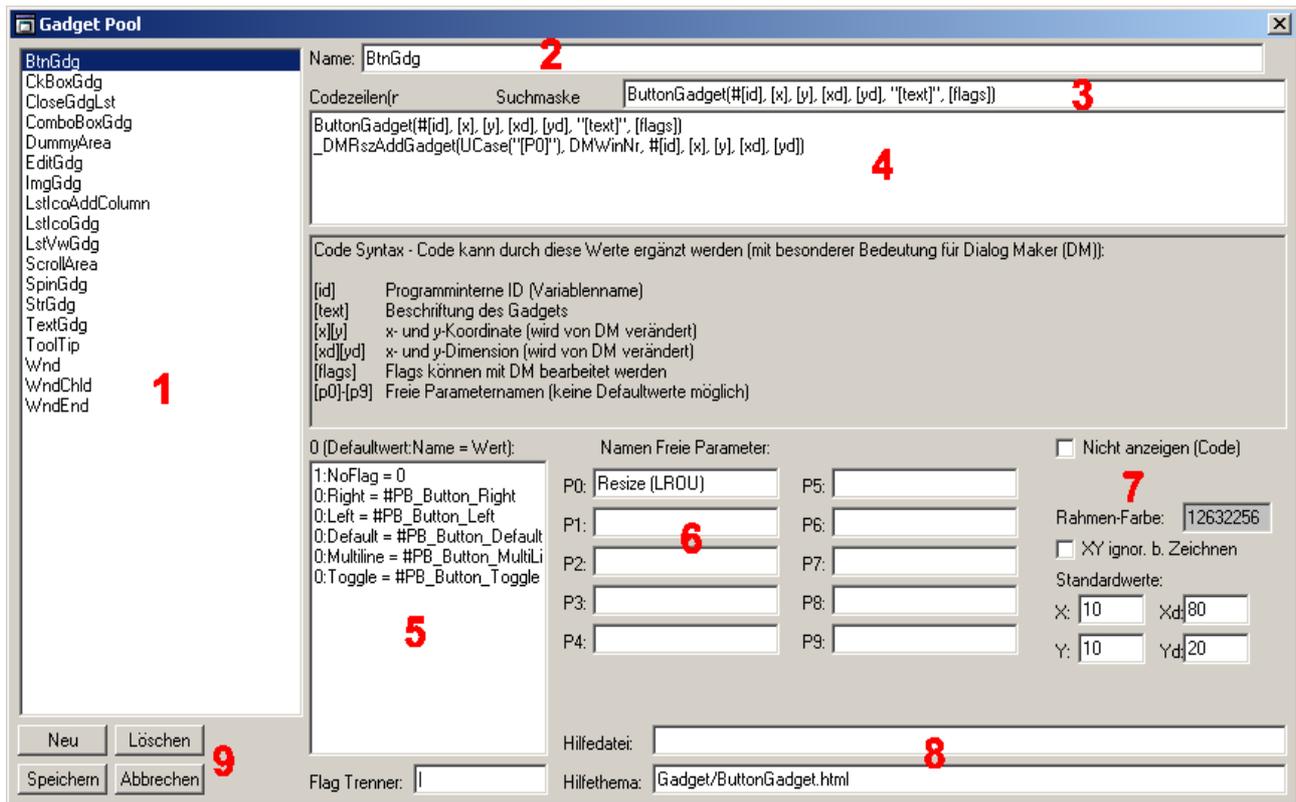
asmsoft@gmx.net

Pool bearbeiten

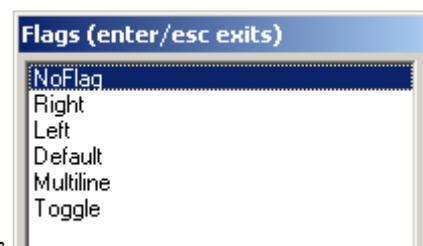
Der Pool enthält die eigentlichen Gadgets, welche im Editor verwendet werden können. Der Pool kann beliebig erweitert werden.

Beim Download von DialogMaker sollte ein Beispiel-Pool beiliegen. Aus dessen Inhalt sollte die Idee grundsätzlich ersichtlich sein... Es sind nicht alle Gadgets enthalten, welche PureBasic zu bieten hat - hauptsächlich jene, welche ich selber schon benötigt habe... :)

Legende



1. Die Liste der verfügbaren Gadgets.
2. Der Name (bzw. Typname) des aktuell ausgewählten Gadgets. Dieser muss einmalig sein!
3. Mit der Suchmaske soll es dereinst möglich sein, bestehende Quelltexte zu scannen und enthaltene Gadgets in DialogMaker zu übernehmen. Diese Funktion ist noch nicht implementiert!
4. Das Codeschnipsel, welches später im Quelltext eingesetzt wird. Darunter einsetzbare Werte (in eckigen Klammern: []). Diese Werte werden von DialogMaker im resultierenden Code entsprechend ersetzt.
5. Hier können mögliche Flags eines Gadgets angegeben werden. Das Beispiel im Bild oben ergibt im Editor eine Auswahl wie im Bild rechts zu sehen. Pro Flag gibt man eine Textzeile ein. Eine Erklärung anhand des Beispiels „1:NoFlag = 0“: „1“ bedeutet, dass dieses Flag standard mässig gesetzt ist,



wenn man ein neues Gadgets von diesem Typ einfügt.

„NoFlag“ ist der Name, welcher in der Liste des Editors angezeigt wird.

„0“ ist der Wert, welcher in den Code eingefügt wird. Dieser Wert wird immer als Zeichenkette interpretiert... (beim zweiten Eintrag zum Beispiel „#PB_Button_Right“).

Werden dann mehrere Flags gleichzeitig ausgewählt, werden dessen Werte verkettet – getrennt durch den „Flag Trenner“ (siehe Eingabefeld unten).

Wird also zum Beispiel „Right“ und „Default“ gewählt würde der erzeugte Code so aussehen: „..., #PB_Button_Right|#PB_Button_Default“.

Der Eintrag „NoFlag“ habe ich im übrigen deshalb erstellt, weil DialogMaker nicht fähig ist, optionale Parameter wegzulassen. Bis jetzt hat dies bei mir als Workaround in PureBasic funktioniert.

6. Namen für zusätzliche Parameter. Diese werden dann im Editor in der Werteliste angezeigt – und es können natürlich auch Werte eingegeben werden (Zeichenketten).
7. Die Option „Nicht anzeigen“ dient dazu, Codeschnipsel erstellen zu können, welche keine Darstellung benötigen (wie zum Beispiel „GadgetToolTip“ in PureBasic), beziehungsweise von DialogMaker nicht dargestellt werden können.
 - „Rahmenfarbe“: In dieser Farbe wird dieses Gadget gezeichnet.
 - „XY ignorieren beim Zeichnen“: Sinnvoll bei Fenster oder ähnlichem, welche von DialogMaker oben rechts dargestellt werden sollen, im erzeugten Code aber durchaus andere Werte haben können.
 - „Standardwerte“: Diese werden beim neu Einfügen im Editor verwendet.
8. Hier kann ein Link auf ein Thema in einer Hilfedatei eingegeben werden. F1 funktioniert aber im Moment nur im Pooleditor. Die Idee wäre, dass die Hilfe zu einem Gadget dann auch mal im Design-Editor mit F1 aufgerufen werden kann.
9. Beenden des Pooleditors – die Funktionen sollten klar sein. Einzig „Neu“ bedarf evtl. der Erklärung: Drückt man darauf wird das aktuelle Gadget kopiert. Das setzt voraus, dass man zumindest den Gadgetnamen ändert – ansonsten erscheint eine Fehlermeldung, dass diese Name schon existiert.

Änderungen werden übernommen, wenn man auf ein anderes Gadget wechselt oder den Pooleditor mit „Speichern“ beendet.

Design Editor

Hier wird das eigentliche Aussehen einer Dialogs bestimmt. Grundsätzlich wählt man in der Gadgetliste (4) einen oder mehrere Einträge. In der Vorschau (1) werden diese rot hervorgehoben.

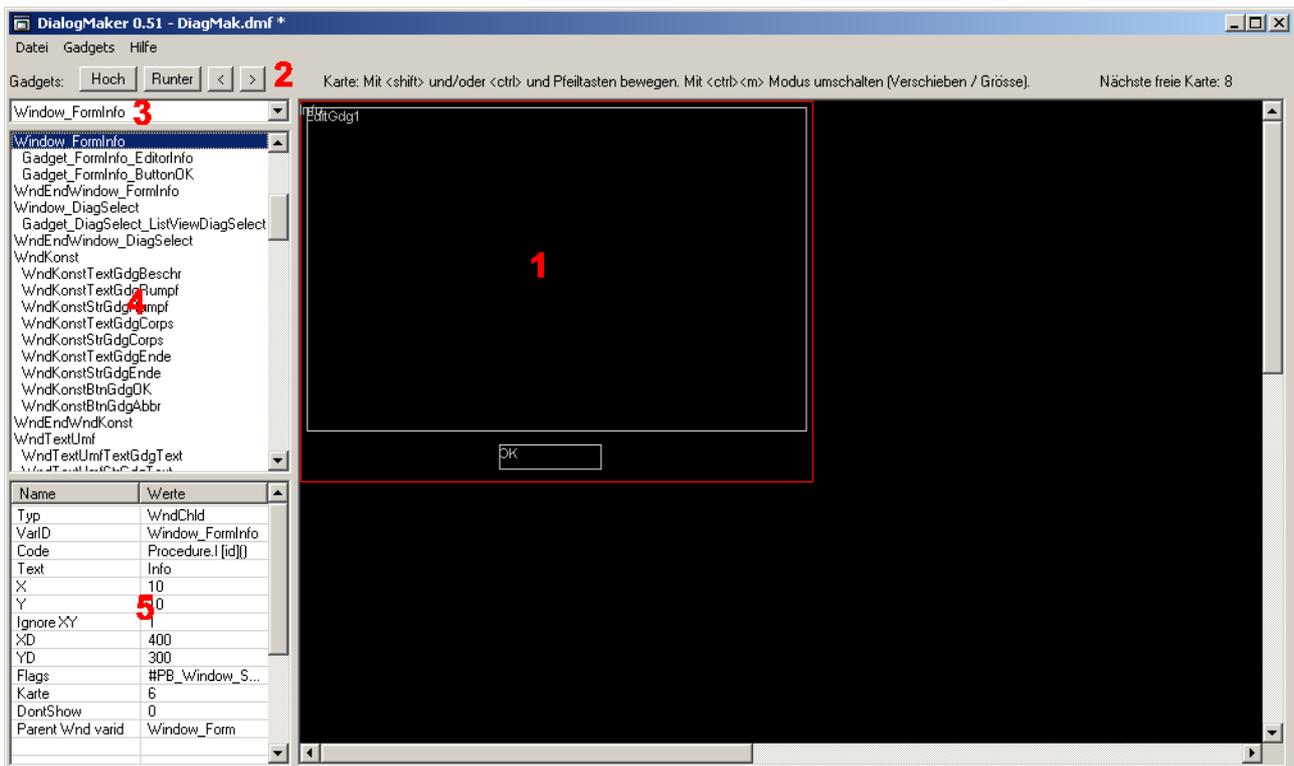
Nun können diese mittels den Pfeiltasten verschoben werden. Dazu muss gleichzeitig folgende Taste(n) gedrückt werden (die Werte können in der DialogMaker.ini geändert werden):

- <SHIFT> um 1 Pixel verschieben
- <CTRL> um 5 Pixel verschieben
- <CTRL><SHIFT> um 50 Pixel verschieben

Wenn man <CTRL><M> drückt, wechselt die Farbe zu Grün, und es kann mit denselben Tastenkombinationen die Größe geändert werden. Nochmaliges <CTRL><M> wechselt wieder auf Rot.

In der Gadgetliste ist es auch möglich, Gadgets zu kopieren, einzufügen und zu löschen (auch mehrere). Dies ist via Menü und Tastaturkürzel möglich.

Legende



1. Die Vorschau – man kann mit der Maus links klicken, um ein einzelnes Gadget auszuwählen.
2. Mit den vier Knöpfen können ein oder mehrere ausgewählte Einträge in der Gadgetliste (4) nach oben, unten, rechts oder links bewegt werden. Die Reihenfolge in der Liste entspricht dem erzeugten Code. Wird ein Gadget nach rechts bewegt, wird es im Code eingerückt (Tabulator).
3. Dient zum schnellen springen zu einer „Karte“.

4. Die Gadgetliste – einen Eintrag anklicken um ihn zu bearbeiten. Alle Einträge, welche dieselbe Kartenummer haben werden in der Vorschau (1) gleichzeitig gezeichnet. Die gewählten in Rot oder Grün, die anderen in Grau.
5. Hier kann man die Werte zu einem Gadget manuell ändern. X, Y, XD und XY werden automatisch geändert, wenn man die Pfeiltasten-Kombinationen verwendet (siehe Einleitung „Design Editor“).
Mit Doppelklick auf einen Wert erscheint ein kleines Eingabefeld in der man die Änderung vornehmen kann. Mit <ENTER> übernimmt man die Eingabe, mit <ESC> verwirft man sie. Es gibt auch Werte, welche nicht geändert werden können (zum Beispiel „Typ“).

Menü

Hier werden nur noch die speziellen Menüpunkte erklärt. Die anderen sollten klar sein...

Menü Gadgets

- **Hinzufügen:** Es wird eine kleine Liste der verfügbaren Gadgets gezeigt. Doppelklick auf einen Eintrag fügt diesen an der aktuelle Position in der Gadgetliste mit den Standardwerten ein.
- **Karte zuordnen:** Ordnet allen ausgewählten Gadgets eine (neue) Kartennummer zu. Es erscheint ein kleiner Dialog, in den man die Nummer eingeben kann.
- **Code zurücksetzen:** Setzt den Code und die Namen von P0-P9 von den gewählten Gadgets zurück auf die Werte des Pools. Dies ist nützlich, wenn man Änderungen vom Pool übernehmen will.
- **Grösseren Text bearbeiten:** Hiermit können grössere, auch mehrzeilige Texte eingegeben werden (für den „Text“-Wert).
- **Varid ersetzen:** Setzt die eindeutigen IDs der Gadgets zurück – es erscheint ein Dialog mit einer Ersetzen-Maske.

