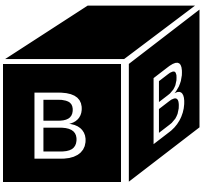


Chapter 5

Dialog Boxes

The design process for programming with dialog

- Decide which elements belong in the dialog box.
- For each input/output element, determine the type of the corresponding variable.
- For each button, determine what processing must be performed.
- Write a module with each of the variables and procedures exported. Compile the module.
- Select Controls→New Form… and link the module to the dialog box.
- Fine tune the layout of the elements of the form.
- Write a Docu document with a commander to activate and test the dialog box.

(a) MacOS.

(b) MSWindows.

Figure 5.1
The dialog box for inputting an integer value shown in mask mode.

```
MODULE Pbox05A;
   IMPORT StdLog;

   VAR
      change*: INTEGER;

   PROCEDURE MakeChange*;
      VAR
         cents: INTEGER;
         dimes, nickels, pennies: INTEGER;
   BEGIN
      cents := change;
      dimes := cents DIV 10;
      cents := cents MOD 10;
      nickels := cents DIV 5;
      pennies := cents MOD 5;
      StdLog.String("You have "); StdLog.Int( change);
      StdLog.String(" cents in change."); StdLog.Ln;
      StdLog.String("Dimes: "); StdLog.Int(dimes); StdLog.Ln;
      StdLog.String("Nickels: "); StdLog.Int(nickels); StdLog.Ln;
      StdLog.String("Pennies: "); StdLog.Int(pennies); StdLog.Ln
   END MakeChange;

BEGIN
   change := 0
END Pbox05A.
```

Figure 5.2
A module for constructing a dialog box to input an integer value.

(a) Before first execution.

(b) Activating dialog box triggers load.

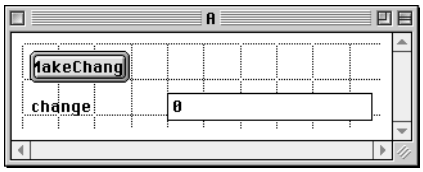
(c) Subsequent executions do not require load.

Figure 5.3
Dynamic loading in the BlackBox framework.

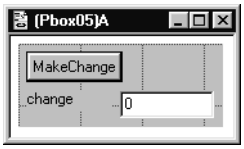
(a) MacOS.

(b) MSWindows.

Figure 5.4
The result of selecting Controls→New Form… .

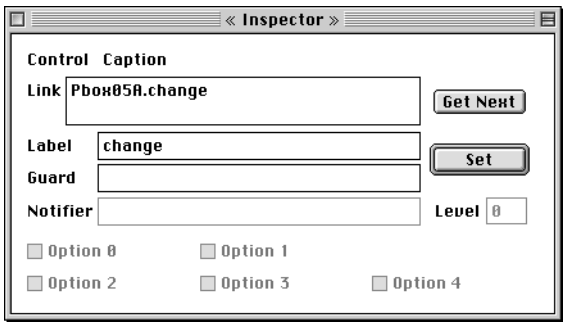


(a) MacOS.

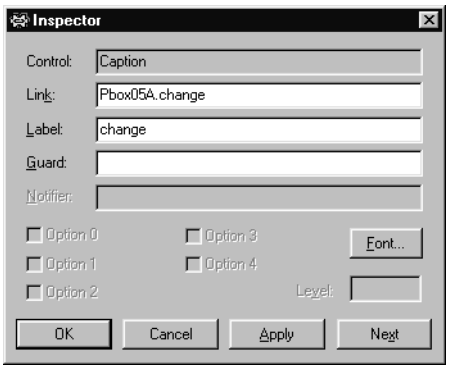


(b) MSWindows.

Figure 5.5
The dialog box shown in layout mode.



(a) MacOS.



(b) MSWindows.

Figure 5.6
The dialog box for editing a control's attributes.

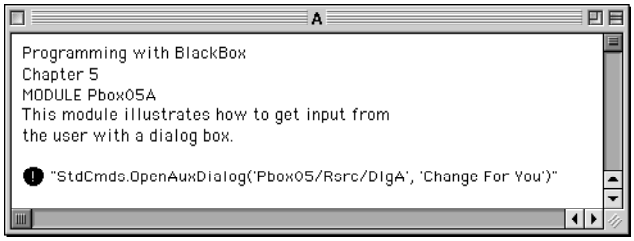
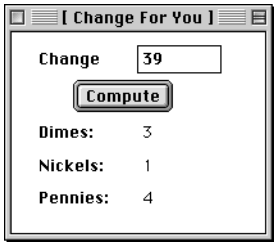
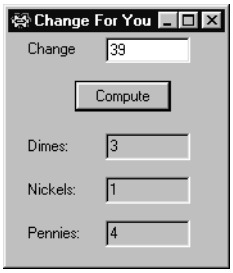


Figure 5.7
The documentation for the module in Listing 5.2



(a) MacOS.

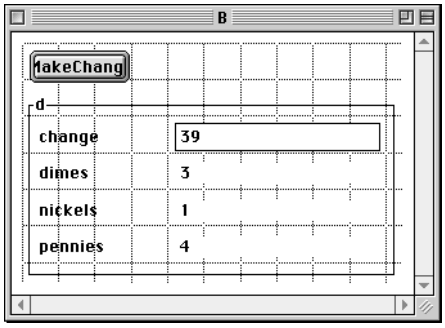


(b) MSWindows.

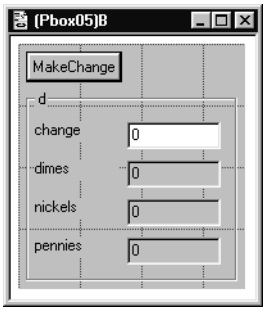
Figure 5.8
A dialog box that displays output as well as input.

```
MODULE Pbox05B;
   IMPORT Dialog;

   VAR
      d*: RECORD
         change*: INTEGER;
         dimes-, nickels-, pennies-: INTEGER
      END;

   PROCEDURE MakeChange*;
      VAR
         cents: INTEGER;
   BEGIN
      cents := d.change;
      d.dimes := cents DIV 10;
      cents := cents MOD 10;
      d.nickels := cents DIV 5;
      d.pennies := cents MOD 5;
      Dialog.Update(d)
   END MakeChange;

BEGIN
   d.change := 0;
   d.dimes := 0; d.nickels := 0; d.pennies := 0
END Pbox05B.
```

Figure 5.9
Sending output to a dialog box.



(a) MacOS.



(b) MSWindows.

Figure 5.10
The default form produced by the forms creator for the module of Figure 5.9.

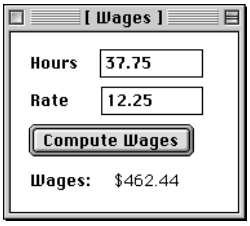


Figure 5.11
A dialog box with string output for Figure 5.12.

```
MODULE Pbox05C;
    IMPORT Dialog, PboxStrings;

    VAR
        d*: RECORD
            hours*, rate*: REAL;
            result-: ARRAY 16 OF CHAR
        END;

    PROCEDURE ComputeWages*;
        VAR
            wage: REAL;
    BEGIN
        wage := d.hours * d.rate;
        PboxStrings.RealToString(wage, 1, 2, d.result);
        d.result :="$" + d.result;
        Dialog.Update(d)
    END ComputeWages;

BEGIN
    d.hours := 0.0; d.rate := 0.0;
    d.result := ""
END Pbox05C.
```

Figure 5.12
A program that produces string output to a dialog box.