

Steps for Creating a Link to an Outside Procedure from within an Object

In what follows the object is assumed to be used by a form - here called the parent form. We'll assume that the object handles the behaviour of a lightswitch and that clicking the lightswitch triggers the operation of the application's business rules starting with a procedure in the parent form.

1. In the object declare a type that takes the form of :
`typename = procedure(Sender : TObject; maybe another parameter) of object;`
Its purpose is to define the nature of the procedure in the parent form that is to be called by the object. Here we'll assume that the type is :
`TOnSwitch = procedure(Sender: TObject; ID : integer) of object;`
2. In the object declare a variable of the type that you declared in step 1. Its purpose is to hold a pointer to the business rules procedure in the parent form Here we'll assume that the variable is :
`FOnSwitch : TOnSwitch;`
3. Also in the object declare a property that is also of the type declared in step 1. having at least a read parameter. Its purpose is to receive a pointer to the business rules procedure in the parent form and assign it to the variable declared in step 2. Here we'll assume that the property is :
`property OnSwitch : TOnSwitch read FOnSwitch write FOnswitch;`
4. Also in the object declare a procedure that is to be triggered by an OnClick event within the object. Its purpose is to send a message telling the parent form to activate the operation of the business rules procedure. The procedure must have a (Sender:TObject) parameter. Here we'll assume that the procedure is :
`procedure SwitchClick (Sender:TObject);`
5. In the object's constructor map the object's OnClick event to the procedure declared in step 4 by using the @ pointer. Here we'll assume that the mapping is :
`OnClick := @SwitchClick;`
Note that only the procedure name is referred to i.e. don't include its parameter.
6. Include in the code for the procedure declared in step 4 a line that takes the following form :
`if Assigned(variable declared in step2) then
 variable declared in step2(Self, maybe another parameter);`
(Remember that the variable declared in step 2 is a pointer to the business rules procedure in the parent form). Here we'll assume that the code line is :
`if Assigned (FOnSwitch) then
 FOnSwitch(Self,num); (num is an object variable)`
7. In the parent form declare a procedure to initiate the operation of the application's business rules. The procedure should have the same pattern of parameters as the type of procedure declared in step 1. Here we'll assume that the declaration is :
`procedure DoOnSwitch(Sender:TObject; ID : integer);`
8. In the parent form (probably in its OnCreate procedure) use the read parameter of the object's property declared in step 3 to point (assign) the variable declared in step 2 to the business rules procedure. Here we'll assume that the assignment is :
`OnSwitch := @DoOnSwitch;` (Again, the procedure's parameters are not included.)
9. In the parent form use the business rule procedure to initiate those business rules. Here we'll assume :
`procedure DoOnSwitch(Sender:TObject; ID : integer)
begin
 flipSwitch(ID); (this starts the business rules running)
end;`

Example of Information Flow for Calling an External Procedure from Within an Object Using the Code Assumed Above

Key :
 Set at design time ——— (green line)
 Operates at run time ——— (blue line)

