How to implement an OnMouseDown event for the buttons of a TRadioGroup

> ***Question***
>
> I have created a decendant of *TRadioGroup* called *TSuperRadioGroup*. With this new control, I have surfaced the *OnMouseDown event*. But the event is only triggered when the mouse goes down on the border or caption of the froup, not the radio buttons themselves.

The solution I use is to register a window procedure for the radio buttons. You can trap the Windows messages there.

```
procedure TSuperRadioGroup.RegisterWndProc;
var
  BtnHnd: hWnd;
  ItrBtn: Integer;
begin
  inherited;
  HasWndProc := True;
  BtnHnd := GetWindow(Handle, GW_CHILD);
  ItrBtn := 0;
  while BtnHnd > 0 do
  begin
    if GetWindowLong(BtnHnd, GWL_USERDATA) <> 0 then
      raise Exception.Create('Userdata may not be used');
    ButtonHandle[ItrBtn] := BtnHnd;
    OrigWndProc[ItrBtn] := GetWindowLong(BtnHnd, GWL_WNDPROC);
    SetWindowLong(BtnHnd, GWL_USERDATA, Longint(self));
    SetWindowLong(BtnHnd, GWL_WNDPROC, Longint(@RadioBtnWndProc));
    Inc(ItrBtn);
    BtnHnd := GetWindow(BtnHnd, GW_HWNDNEXT);
  end;
end;
```

In the *RadioBtnWndProc* window procedure you can use this code to get at the radiogroup object and the specific button:

```
Obj := TObject(GetWindowLong(WndHnd, GWL_USERDATA));
if Obj is TSuperRadioGroup then
begin
  RadioGrp := TSuperRadioGroup(Obj);
  for ItrBtn := 0 to RadioGrp.Items.Count - 1 do
  begin
    if WndHnd = RadioGrp.ButtonHandle[ItrBtn] then
    begin
      OrigWndProc := RadioGrp.OrigWndProc[ItrBtn];
      break;
    end;
  end;
end;
```

If the message is not completely handled, you need to call the original window procedure at the end of your specialized window procedure:

```
Result := CallWindowProc(
  Pointer(OrigWndProc), WndHnd, Msg, WParam, LParam)
;
```

Original resource:	The Delphi Pool
Author:	Unknown
Added:	2009-10-28
Last updated:	2009-10-28