Update: Custom Clipboard Formats

by Xavier Pacheco

Thanks to Hallvard Vassbotn for pointing out an oversight in Issue 3's *Custom Clipboard Formats* article. In the article, the TBirthDay.PasteFromClipBoard method declared a variable Size, which was used in the statement:

```
if SizeOf(FPersonRec)  GlobalSize(Data) then
  Size := GlobalSize(Data);
```

however, the following statement neglected to make use of the Size variable:

```
Move(DataPtr^, FPersonRec, SizeOf(FPersonRec));
```

The correct code should read:

```
if SizeOf(FPersonRec)  GlobalSize(Data) then
  Size := GlobalSize(Data)
else
  Size := SizeOf(FPersonRec);
Move(DataPtr^, FPersonRec, Size);
```

Also, the outer try..except statement in that same method was unnecessary. The memory referenced by the Data variable does not have to be freed in this method since Windows will manage this memory block already. Listing 1 below shows the corrected version of the PasteFromClipBoard method (also on the disk).

➤ *Listing 1*

```
procedure TBirthday.PasteFromClipBoard;
var
  Data: THandle;
  DataPtr: Pointer;
  Size: Integer;
begin
  { Get the data on the clipboard }
  Data := ClipBoard.GetAsHandle(CF_BIRTHDAY);
  { Exit is unsuccessful }
  if Data = 0 then Exit;
  { Lock the Global memory object }
  DataPtr := GlobalLock(Data);
  try
    if SizeOf(FPersonRec) > GlobalSize(Data) then
      Size := GlobalSize(Data)
    else
      Size := SizeOf(FPersonRec);
    { Copy contents of DataPtr to Buffer }
    Move(DataPtr^, FPersonRec, Size);
  finally
    GlobalUnlock(Data); {Unlock global memory object}
  end;
end;
```