# TDataSet Update

In the September and October issues, we developed a customize TDataSet descendant component to allow us to access a proprietary database format. I was fortunate enough to have Borland's Mark Edington, the engineer responsible for the TDataSet abstraction in Delphi 3, take a look at those articles. Mark pointed out a couple of flaws in my implementation.

First, the CompareBookmarks method should return -1 if Bookmark1 refers to a location before Bookmark2 in the table (contrary to what the online help says). This behavior is needed by TDBGrid when multiselect is enabled. Listing 3 shows Mark's corrected method.

Second, in the GetRecord method, a problem exists when the last physical record of the file has been deleted. In my original code, an infinite loop would occur as the dataset tried to call for the prior record from the end of the file. Listing 4 shows the corrected code for the gmPrior case within the GetRecord method.

```
function TMyDataSet.CompareBookmarks(
  Bookmark1, Bookmark2: TBookmark): Integer;
const
  RetCodes: array[Boolean, Boolean]
    of Integer = ((2, -1), (1, 0));
begin
  Result := RetCodes[Bookmark1 = nil, Bookmark2 = nil];
  if Result = 2 then begin
    if TBookmarkInfo(Bookmark1^) =
      TBookmarkInfo(Bookmark2^) then
      Result := 0
    else if TBookmarkInfo(Bookmark1^) >
      TBookmarkInfo(Bookmark2^) then
      Result := 1
    else
      Result := -1;
  end;
end;
```

➤ *Listing 3*

```
gmPrior:
  begin
    AtEof := Eof;
    repeat
      FilePosition := FilePos(FInternalFile);
      if FilePosition < (2 * FRecSize) then
        Result := grBOF
      else begin
        if AtEof then
          Seek(FInternalFile,
            FileSize(FInternalFile) - FRecSize)
        else
          Seek(FInternalFile,
            FilePosition - (2 * FRecSize));
        BlockRead(FInternalFile, Buffer^, FRecSize);
        AtEof := False;
      end;
    until (Result <> grOk) or (Byte(Buffer^) = 0);
  end;
```

➤ *Listing 4*