

```
1 Extend the Firemonkey TGrid with MultiSelect
2 By M.C. van der Kooij
3 www.PictoSelector.eu
4
5 Copy the following file from
6 C:\Program Files (x86)\Embarcadero\RAD Studio\11.0\source\fm\FMX.Grid.pas
7 to your own project folder
8
9 Now add this copied file to your project.
10
11 Add to the first uses:
12     System.Generics.Collections,
13
14 Add after TCustomGrid = class;
15
16     TMouseAction = (maDown, maMove, maUp);
17
18     TUniqueList<T> = class(TList<T>)
19     public
20         function Add(const Value: T): Integer;
21     end;
22
23
24 In TCustomGrid interface:
25     Add to private:
26         FMultiSelect: boolean;
27
28     Replace methods in private:
29         procedure SelCell(Col, Row: Integer; Shift: TShiftState);
30         procedure InternalClick(X, Y: Single; Shift: TShiftState; Action: TMouseActi
31
32     Add to private:
33         procedure SetMultiSelect(const Value: boolean);
34
35     Add to protected:
36         FSelectedRows: TUniqueList<integer>;
37         function GetSelectedRows: TUniqueList<integer>;
38
39     Add to public (just after the declaration of the last property):
40         property SelectedRows: TUniqueList<integer> read GetSelectedRows;
41         property MultiSelect: boolean read FMultiSelect write SetMultiSelect;
42
43 In TGrid interface:
44     Add to published:
45         property MultiSelect default False;
46
47
48 In constructor TCustomGrid.Create(AOwner: TComponent);
49     Add after FRowCount := 100;
50         FSelectedRows := TUniqueList<integer>.Create;
51         FMultiSelect := False;
52
53 In destructor TCustomGrid.Destroy;
54     Add after begin
55         if Assigned(FSelectedRows) then
56             FreeAndNil(FSelectedRows);
57
58 In procedure TCustomGrid.KeyDown(var Key: Word; var KeyChar: System.WideChar; Shift: TShiftS
59     Add after // Change selection
60     And before Selected := LSelected
61         if not FMultiSelect or not (ssCtrl in Shift) then begin
62             FSelectedRows.Clear;
63         end;
64         FSelectedRows.Add(LSelected);
65
66 Delete the method: procedure TCustomGrid.SelCell(Col, Row: Integer);
67
68 Replace the line: procedure TCustomGrid.InternalClick(X, Y: Single);
69     with:
70         procedure TCustomGrid.InternalClick(X, Y: Single; Shift: TShiftState; Action
71
72     and replace the lines:
73         // Change selection
74         SelCell(NewCol, NewRow);
75         // Observers
76         if ChangeRow then
```

```
77             TLinkObservers.PositionLinkPosChanged(Observers);
78
79             end;
80         end;
81     end;
82     with:
83         // Change selection
84         SelCell(NewCol, NewRow, Shift);
85         // Observers
86         if ChangeRow then
87             TLinkObservers.PositionLinkPosChanged(Observers);
88
89         end else if Assigned(Column) and (Action = maDown) then begin
90             // Change selection
91             SelCell(NewCol, NewRow, Shift);
92         end;
93     end;
94 end;
95
96 Replace in procedure TCustomGrid.MouseDown(Button: TMouseButton; Shift: TShiftState; X, Y: S
97 the line:
98     InternalClick(X, Y);
99     with:
100     InternalClick(X, Y, Shift, maDown);
101
102 Replace in procedure TCustomGrid.MouseUp(Button: TMouseButton; Shift: TShiftState; X, Y: Sin
103 the line:
104     InternalClick(X, Y);
105     with:
106     InternalClick(X, Y, Shift, maUp);
107
108 Replace in procedure TCustomGrid.MouseMove(Shift: TShiftState; X, Y: Single);
109 the line:
110     InternalClick(X, Y);
111     with:
112     InternalClick(X, Y, Shift, maMove);
113
114 Add to procedure TCustomGrid.SetRowCount(const Value: Integer);
115 after the lines with:
116     if (FSelected >= FRowCount) and (FRowCount > 0) then begin
117         Selected := FRowCount - 1;
118     the line:
119         FSelectedRows.Clear;
120         FSelectedRows.Add(Selected);
121
122 Replace in function TCustomGrid.IsSelected(Row: Integer): Boolean;
123 the line:
124     Result := (Row = FSelected);
125     with:
126     Result := FSelectedRows.Contains(Row);
127
128
129
130
131
132 Add the following code before the initialization:
133
134 function TCustomGrid.GetSelectedRows: TUniqueList<integer>;
135 begin
136     Result := FSelectedRows;
137 end;
138
139 procedure TCustomGrid.SelCell(Col, Row: Integer; Shift: TShiftState);
140 var
141     iPreviousSelected, index: integer;
142 begin
143     if (Col <> ColumnIndex) or (Row <> Selected) then
144     begin
145         if (FFocus <> nil) then
146             FFocus.Visible := False;
147         BeginUpdate;
148         try
149             Root.SetActiveControl(nil);
150             if (ColumnIndex > 0) and (ColumnIndex < ColumnCount) then
151                 begin
152                     Columns[ColumnIndex].FEditMode := -1;
```

```
153     end;
154   finally
155     EndUpdate;
156   end;
157   ColumnIndex := Col;
158 end;
159   iPreviousSelected := Selected;
160
161   if not FMultiSelect or not ((ssShift in Shift) or (ssCtrl in Shift)) then begin
162     FSelectedRows.Clear;
163     FSelectedRows.Add(Row);
164   end else begin
165     if FSelectedRows.Contains(Row) then begin
166       FSelectedRows.Remove(Row);
167     end else begin
168       FSelectedRows.Add(Row);
169     end;
170   end;
171
172   if (ssShift in Shift) and (iPreviousSelected >= 0) and FMultiSelect then begin
173     for index := Min(iPreviousSelected, Row) to Max(iPreviousSelected, Row) do begin
174       FSelectedRows.Add(index);
175     end;
176   end;
177
178   Selected := Row;
179 end;
180
181 procedure TCustomGrid.SetMultiSelect(const Value: boolean);
182 begin
183   if MultiSelect <> Value then begin
184     FMultiSelect := Value;
185     if not FMultiSelect then begin
186       FSelectedRows.Clear;
187       FSelectedRows.Add(Selected);
188       RealignContent;
189     end;
190   end;
191 end;
192
193 { TUniqueList<T> }
194
195 function TUniqueList<T>.Add(const Value: T): Integer;
196 begin
197   if not Contains(Value) then
198     Result := inherited Add(Value);
199 end;
200
```