

```

1: unit HUDirNameEntry;
2:
3: {$mode objfpc}{$H+}
4:
5: //=====
6: //
7: // Unit : HUDirNameEntry.pas
8: //
9: // Description :
10: //
11: // Called By :
12: //
13: // Calls :   HUConstants
14: //           HUMessageBoxes : HUConfirmMsgYN
15: //           HUValidations : ValidDirectoryCharacter
16: //
17: // Ver. : 1.0.0
18: //
19: // Date : 20 Jan 2019
20: //
21: //=====
22:
23: interface
24:
25: uses
26:   Buttons, Classes, SysUtils, FileUtil, Forms, Controls, Graphics, Dialogs, StdCtrls,
27:   // Application units
28:   AppSettings,
29:   // HULibrary units
30:   HUConstants, HUMessageBoxes, HUValidations;
31:
32: type
33:
34:   { TdlgHUDirNameEntry }
35:
36:   TdlgHUDirNameEntry = class(TForm)
37:     bbtCancel: TBitBtn;
38:     bbtOK: TBitBtn;
39:     edtDirName: TEdit;
40:     procedure bbtCancelClick(Sender: TObject);
41:     procedure bbtOKClick(Sender: TObject);
42:     procedure edtDirNameKeyPress(Sender: TObject; var Key: char);
43:     procedure edtDirNameKeyUp(Sender: TObject; var Key: Word; Shift: TShiftState );
44:     procedure FormCreate(Sender: TObject);
45:     procedure FormShow(Sender: TObject);
46:   private
47:     vstrOriginalDirName : string;
48:     fDirName : string;
49:     fDirPath : string;
50:     function GetDirName : string;
51:     procedure SetDirName(Dir : string);
52:     function GetDirPath : string;
53:     procedure SetDirPath(DirPath : string);
54:   public
55:     property pDirName : string read GetDirName write SetDirName;
56:     property pDirPath : string read GetDirPath write SetDirPath;
57:   end; // TdlgHUDirNameEntry
58:
59: var
60:   dlgHUDirNameEntry: TdlgHUDirNameEntry;

```

```

61:
62: implementation
63:
64: {$R *.lfm}
65:
66: //=====
67: //          PRIVATE CONSTANTS
68: //=====
69: const
70:   cintMinDirNameCharacters = 1;
71:   cintMaxDirNameCharacters = 20;
72:
73: //=====
74: //          PUBLIC CONSTANTS
75: //=====
76:
77: //=====
78: //          PRIVATE VARIABLES
79: //=====
80:
81: //=====
82: //          PUBLIC VARIABLES
83: //=====
84:
85: //=====
86: //          PRIVATE ROUTINES
87: //=====
88:
89: //=====
90: //          PUBLIC ROUTINES
91: //=====
92:
93: //=====
94: //          PROPERTY ROUTINES
95: //=====
96: function TdlgHUDirNameEntry.GetDirName: string;
97: begin
98:   Result := fDirName;
99: end; // function TdlgHUDirNameEntry.GetDirName
100:
101: //-----
102: procedure TdlgHUDirNameEntry.SetDirName(Dir: string);
103: begin
104:   fDirName := Dir;
105: end; // procedure TdlgHUDirNameEntry.SetDirName
106:
107: //=====
108: function TdlgHUDirNameEntry.GetDirPath: string;
109: begin
110:   Result := fDirPath;
111: end; // function TdlgHUDirNameEntry.GetDirPath
112:
113: //-----
114: procedure TdlgHUDirNameEntry.SetDirPath(DirPath: string);
115: begin
116:   fDirPath := DirPath;
117: end; // procedure TdlgHUDirNameEntry.SetDirPath
118:
119: //=====
120: //          MENU ROUTINES

```

```

121: //=====
122:
123: //=====
124: //          COMMAND BUTTON ROUTINES
125: //=====
126:
127: //=====
128: //          CONTROL ROUTINES
129: //=====
130: procedure TdlgHUIDirNameEntry.bbtCancelClick(Sender: TObject);
131: begin
132:   if HUConfirmMsgYN('Confirm', 'Do you really want to Cancel ?') = mrNo then
133:   begin
134:     edtDirName.SetFocus;
135:     ModalResult := mrNone;
136:     Exit;
137:   end;
138:
139:   ModalResult := mrCancel;
140:
141: end; // procedure TdlgHUIDirNameEntry.bbtCancelClick
142:
143: //-----
144: procedure TdlgHUIDirNameEntry.bbtOKClick(Sender: TObject);
145: var
146:   vstrDirFullPath : string;
147: begin
148:
149:   if edtDirName.Text = '' then
150:   begin
151:     showmessage('You Must Enter a Valid Directory Name');
152:     edtDirName.SetFocus;
153:     ModalResult := mrNone;
154:     Exit;
155:   end; // if edtDirName.Text = ''
156:
157:   pDirName := edtDirName.Text;
158:
159:   // Confirm Dir DOES NOT already exist
160:   pDirPath := pDirPath + '\' + pDirName;
161:
162:   If DirectoryExists(pDirPath) then
163:   begin
164:     showmessage('Directory ' + pDirPath + ' Already Exists!');
165:     edtDirName.Text := '';
166:     edtDirName.SetFocus;
167:     ModalResult := mrNone;
168:     Exit;
169:   end; // If DirectoryExists(pDirPath)
170:
171: end; // procedure TdlgHUIDirNameEntry.bbtOKClick
172:
173: //=====
174: procedure TdlgHUIDirNameEntry.edtDirNameKeyPress(Sender: TObject; var Key: char);
175: begin
176:   Key := HUValidations.ValidDirectoryCharacter(Key);
177: end; // procedure TdlgHUIDirNameEntry.edtDirNameKeyPress
178:
179: //-----
180: procedure TdlgHUIDirNameEntry.edtDirNameKeyUp(Sender: TObject; var Key: Word;

```

```
181:     Shift: TShiftState);
182: begin
183:     if Length(edtDirName.Text) < cintMinDirNameCharacters then
184:         edtDirName.Color := bclrError
185:     else
186:         edtDirName.Color := bclrOk;
187: end;// procedure TdlgHUDirNameEntry.edtDirNameKeyUp
188:
189: //=====
190: //          FILE ROUTINES
191: //=====
192:
193: //=====
194: //          FORM ROUTINES
195: //=====
196: procedure TdlgHUDirNameEntry.FormCreate(Sender: TObject);
197: begin
198:     edtDirName.MaxLength := cintMaxDirNameCharacters;
199: end;// procedure TdlgHUDirNameEntry.FormCreate
200:
201: //=====
202: procedure TdlgHUDirNameEntry.FormShow(Sender: TObject);
203: begin
204:
205:     edtDirName.Text := PDirName;
206:
207:     if Length(edtDirName.Text) < cintMinDirNameCharacters then
208:         edtDirName.Color := bclrError
209:     else
210:         edtDirName.Color := bclrOk;
211:
212:     edtDirName.SetFocus;
213:
214: end;// procedure TdlgHUDirNameEntry.FormShow
215:
216: //=====
217: end.// unit HUDirNameEntry
218:
```