

```

1: unit HUValidations;
2:
3: {$mode objfpc}{$H+}
4:
5: //=====
6: //
7: // Unit : HUValidations.pas
8: //
9: // Description :
10: //
11: // Called By :
12: //
13: // Calls :
14: //
15: // Ver. : 1.0.0
16: //
17: // Date : 10 Feb 2019
18: //
19: //=====
20:
21: interface
22:
23: uses
24:   Classes, SysUtils, Dialogs,
25:   // Application units
26:   // HULib units
27:   HUConstants;
28:
29:
30: //function ValidAlphaCharacter( Key: char) : char;
31: function ValidNameCharacter( Key: char) : char;
32: function ValidCallsignCharacter( Key: char) : char;
33: function ValidEmailCharacter( Key: char) : char;
34: function ValidDirectoryCharacter( Key: char) : char;
35: function ValidDigitCharacter( Key: char) : char;
36:
37: implementation
38:
39: //=====
40: //          PRIVATE CONSTANTS
41: //=====
42:
43: //=====
44: //          PUBLIC CONSTANTS
45: //=====
46:
47: //=====
48: //          PRIVATE VARIABLES
49: //=====
50:
51: //=====
52: //          PUBLIC VARIABLES
53: //=====
54:
55: //=====
56: //          PRIVATE ROUTINES
57: //=====
58:
59: //=====
60: //          PUBLIC ROUTINES

```

```

61: //=====
62: {function ValidAlphaNumericCharacter( Key: char) : char;
63: begin
64:     // Returns only Valid Alphabetic Characters. Non-valid characters are converted
65:     // into Null (#0) characters.
66:     //Valid Alpha C haracters are:
67:     // <BS>
68:     // <SP>
69:     // [A..Z]
70:     // [a..z]
71:     // [0..9]
72:     Result := Key;
73:     case Key of
74:         keyBS : Exit; // <BS>
75:         keySpace : Exit; // <SP>
76:         key0..key9 : Exit; // [0..9]
77:         keyA..keyZ : Exit; // [A..Z]
78:         key_a..key_z : Exit; // [a..z]
79:     else
80:         Result := keyNull;
81:     end; // case Key of
82: end; // function ValidAlphaNumericCharacter(var Key: char);
83:
84: //=====
85: function ValidCallsignCharacter( Key: char) : char;
86: begin
87:     // Returns only Valid Callsign Characters. Non-valid characters are converted
88:     // into Null (#0) characters.
89:     //Valid Alpha C haracters are:
90:     // <BS>
91:     // </>
92:     // [0..9]
93:     // [A..Z]
94:     // [a..z] Converted to Uppercase
95:     Result := Key;
96:     case Key of
97:         K_BS : Exit; // <BS>
98:         K_ForwardSlash : Exit; // </>
99:         K_0..K_9 : Exit; // [0..9]
100:        uK_A..uK_Z : Exit; // [A..Z]
101:        lK_a..lK_z : begin
102:            Result := UpCase(Key);
103:            Exit; // [a..z]
104:        end;
105:     else
106:         Result := K_NULL;
107:     end; // case Key of
108:
109: end; // function ValidCallsignCharacter(var Key: char);
110:
111: //=====
112: function ValidDigitCharacter( Key: char) : char;
113: begin
114:     // Returns only Valid Digits. Non-valid characters are converted
115:     // into Null (#0) characters.
116:     //Valid Digit Characters are:
117:     // <BS>
118:     // <DEL>
119:     // [0..9]
120:

```

```

121: Result := Key;
122: case Key of
123:     K_BS : Exit; // <BS>
124:     K_0..K_9 : Exit; // [0..9]
125: else
126:     Result := K_NULL;
127: end; // case Key of
128:
129: end; // function ValidDigitCharacter(var Key: char);
130:
131: //=====
132: {Function ValidFloatCharacter( Key: char) : char;
133: begin
134:     // Returns only Valid Digits and the Decimal Point. Non-valid characters are converted
135:     // into Null (keyNull) characters.
136:     // Valid Digit Characters are:
137:     // <BS>
138:     // [.]
139:     // <DEL>
140:     // [0..9]
141:     Result := Key;
142:     case Key of
143:         keyBS : Exit; // <BS>
144:         keyDecimalPoint : Exit; // <.>
145:         keyDEL : Exit; // <DEL>
146:         key0..key9 : Exit; // [0..9]
147:     else
148:         Result := keyNull;
149:     end; // case Key of
150: end; // Function ValidFloatCharacter( Key: char)}
151:
152: //=====
153: {function ValidFilenameCharacter(Key: char) : char;
154: begin
155:     // Returns only Valid Filename Characters. Non-valid characters are converted
156:     // into Null (#0) characters.
157:     //Valid Alpha C haracters are:
158:     // <BS>
159:     // <SP>
160:     // [A..Z]
161:     // [a..z]
162:     // [0..9]
163:     // <_>
164:     Result := Key;
165:     case Key of
166:         keyBS : Exit; // <BS>
167:         keySpace : Exit; // <SP>
168:         key0..key9 : Exit; // [0..9]
169:         keyA..keyZ : Exit; // [A..Z]
170:         keyUScore : Exit; // <_>
171:         key_a..key_z : Exit; // [a..z]
172:     else
173:         Result := keyNull;
174:     end; // case Key of
175: end; // function ValidFilenameCharacter }
176:
177: //=====
178: function ValidDirectoryCharacter (Key: char) : char;
179: begin
180:

```

```

181:      // Returns only Valid Directory Characters. Non-valid characters are converted
182:      // into Null (#0) characters.
183:      //Valid Alpha Characters are:
184:      // <BS>
185:      // <SP>
186:      // [A..Z]
187:      // [a..z]
188:      // [0..9]
189:      // <_>
190:      Result := Key;
191:      case Key of
192:          K_BS : Exit; // <BS>
193:          K_SP : Exit; // <SP>
194:          K_0..K_9 : Exit; // [0..9]
195:          uK_A..uK_Z : Exit; // [A..Z]
196:          k_UnderScore : Exit; // <_>
197:          lK_a..lK_z : Exit; // [a..z]
198:      else
199:          Result := K_NULL;
200:      end; // case Key of
201:
202: end; // function ValidDirectoryCharacter
203:
204: //=====
205: function ValidNameCharacter (Key: char) : char;
206: begin
207:
208:     // Returns only Valid Name Characters. Non-valid characters are converted
209:     // into Null (#0) characters.
210:     //Valid Name Characters are:
211:     // <BS>
212:     // <SP>
213:     // [A..Z]
214:     // [a..z]
215:     // <->
216:     Result := Key;
217:     case Key of
218:         K_BS : Exit; // <BS>
219:         K_SP : Exit; // <SP>
220:         uK_A..uK_Z : Exit; // [A..Z]
221:         lK_a..lK_z : Exit; // [a..z]
222:         K_UnderScore : Exit; // <_>
223:     else
224:         Result := K_NULL;
225:     end; // case Key of
226:
227: end; // function ValidName Characters
228:
229: //=====
230: function ValidEmailCharacter (Key: char) : char;
231: begin
232:
233:     // Returns only Valid Name Characters. Non-valid characters are converted
234:     // into Null (#0) characters.
235:     //Valid Name Characters are:
236:     // <BS>
237:     // <SP>
238:     // [A..Z]
239:     // [a..z]
240:     // <->

```

```
241: Result := Key;
242: case Key of
243:   K_BS : Exit; // <BS>
244:   K_SP : Exit; // <SP>
245:   uK_A..uK_Z : Exit; // [A..Z]
246:   lK_a..lK_z : Exit; // [a..z]
247:   K_0..K_9 : Exit; // [0..9]
248:   K_ExclamationMark : Exit; // !
249:   K_HashMark : Exit; // #
250:   K_DollarSign : Exit; // &
251:   K_PercentSign : Exit; // %
252:   K_Ampersand : Exit; // &
253:   K_SingleQuote : Exit; // '
254:   K_Asterix : Exit; // *
255:   K_PlusSign : Exit; // +
256:   K_ForwardSlash : Exit; // /
257:   K_EqualSign : Exit; // =
258:   K_QuestionMark : Exit; // ?
259:   K_Caret : Exit; // ^
260:   K_UnderScore : Exit; // _
261:   K_RightSingleQuote : Exit; // '
262:   K_LeftBracket : Exit; // {
263:   K_VerticalBar : Exit;
264:   K_RightBracket : Exit; // }
265:   K_Tilde : Exit; // ~
266:   K_Dash : Exit; // -
267: else
268:   Result := K_NULL;
269: end; // case Key of
270:
271: end; // function ValidEmailCharacter
272:
273: //=====
274: //          PROPERTY ROUTINES
275: //=====
276:
277: //=====
278: //          MENU ROUTINES
279: //=====
280:
281: //=====
282: //          FILE ROUTINES
283: //=====
284:
285: //=====
286:
287: end. // unit HUValidations
288:
```