

```

1: unit Register;
2:
3: //=====
4: //
5: //   Register.pas
6: //
7: //
8: //   Calls: AppVariables
9: //           HUtils : CalculateRegistrationKey
10: //                  ErrorMessageDlgOk
11: //                  ValidAlphaCharacter
12: //                  ValidCallsignCharacter
13: //                  ValidDigitCharacter
14: //           Main : TfrmMain.DisplayTitleBar
15: //
16: //   Called By: INIStuff : ReadINIFile
17: //                  WriteINIFile
18: //           NagScreen : TdlgNagScreen.bbtRegisterClick
19: //
20: //   Ver: 1.0.0
21: //
22: //   Date: 21 Dec 2013
23: //
24: //=====
25:
26: {$mode objfpc}{$H+}
27:
28: interface
29:
30: uses
31:   Classes, SysUtils, FileUtil, Forms, Controls, Graphics, Dialogs, StdCtrls,
32:   Buttons,
33:   // Application Units
34:   AppVariables, HUtils;
35:
36: type
37:
38:   TdlgRegister = class(TForm)
39:     bbtOk: TBitBtn;
40:     bbtCancel: TBitBtn;
41:     bbtHelp: TBitBtn;
42:     edtName: TEdit;
43:     edtCallsign: TEdit;
44:     edtRegKey: TEdit;
45:     Label1: TLabel;
46:     Label2: TLabel;
47:     Label3: TLabel;
48:     Label4: TLabel;
49:     procedure bbtCancelClick(Sender: TObject);
50:     procedure bbtOkClick(Sender: TObject);
51:     procedure edtCallsignKeyPress(Sender: TObject; var Key: char);
52:     procedure edtNameKeyPress(Sender: TObject; var Key: char);
53:     procedure edtRegKeyKeyPress(Sender: TObject; var Key: char);
54:     procedure FormShow(Sender: TObject);
55:   private
56:     { private declarations }
57:   public
58:     { public declarations }
59:   end;
60:

```

```

61: var
62:   dlgRegister: TdlgRegister;
63:
64: implementation
65:
66: {$R *.lfm}
67:
68: uses
69:   Main;
70:
71: const
72:   cbytMaxNameLength = 20;
73:   cbytMinNameLength = 2;
74:   cbytMaxCallsignLength = 10;
75:   cbytMinCallsignLength = 3;
76:   cbytRegKeyLength = 8;
77:
78:   cstrNameEntry =      '      You must enter a Valid Name.';
79:   cstrValidName =      '      A Valid Name consists of two to 20' +
80:                         #13 +
81:                         '      Alphabetic characters and a space.';
82:   cstrCallsignEntry =  '      If entered, your Callsign must be valid.';
83:   cstrValidCallsign =  '      A Valid Callsign consists of three to 11' +
84:                         #13 +
85:                         '      Alphabetic and Numeric characters and the' +
86:                         #13 +
87:                         '      / character.';
88:   cstrRegKeyEntry =    '      You must enter a Valid Registration Key.';
89:   cstrValidRegKey =    '      A Valid Registration Key consists of eight' +
90:                         #13 +
91:                         '      Digits [0..9].';
92:   cstrUnmatchedRegKey = 'The Registration Key entered is not a correct' +
93:                         #13 +
94:                         ' match for the Name and/or Callsign entered.';
95:   cstrCancelButtonHint = 'Cancel the Registration Process.';
96:   cstrOKButtonHint = 'Complete the Registration Process.';
97:
98: //=====
99: //          FORM ROUTINES
100: //=====
101: procedure TdlgRegister.FormShow(Sender: TObject);
102: begin
103:   dlgRegister.Caption := 'TMV7A Registration';
104:   edtName.Hint := cstrValidName;
105:   edtName.MaxLength := cbytMaxNameLength;
106:   edtName.Text := gvstrRegisteredName;
107:   edtCallsign.Hint := cstrValidCallsign;
108:   edtCallsign.MaxLength := cbytMaxCallsignLength;
109:   edtCallsign.Text := gvstrRegisteredCall;
110:   edtRegKey.Hint := cstrValidRegKey;
111:   edtRegKey.MaxLength := cbytRegKeyLength;
112:   edtRegKey.Text := gvstrRegistrationKey;
113:   bbtCancel.Hint := cstrCancelButtonHint;
114:   bbtOK.Hint := cstrOKButtonHint;
115:   edtName.SetFocus;
116: end;
117:
118: //=====
119: //          BUTTON ROUTINES
120: //=====

```

```

121: procedure TdlgRegister.bbtOkClick(Sender: TObject);
122:
123: var
124:     vstrRegString : string;
125:
126: begin
127:
128:     // We must have valid data in the Name and optionally Callsign fields as well as the
129:     // Registration Key field. If there is a Callsign, then it will be used to calculate
130:     // The Registration Key Value. If there is no Callsign, the Name will be used. The
131:     // Calculated value will be compared to the value entered into the Registration Key
132:     // field to determine Registration Validation.
133:
134:     // First we check the validity of the fields. Any non-valid data entries will
135:     // cause an error message to be displayed and the form will remain open.
136:
137:     // We require at least one of the Name or Callsign fields to calculate the
138:     // Registration Key
139:
140:     // The name must be Alphabetic with 2 to 20 characters
141:     if (Length(edtName.Text) < cbytMinNameLength) then
142:     begin
143:         ErrorMessageDlgOk('Invalid Name', cstrNameEntry +
144:                             #13 +
145:                             #13 +
146:                             cstrValidName);
147:         ModalResult := mrNone;
148:         edtName.SetFocus;
149:         Exit;
150:     end; // if (Length(edtName.Text) < cbytMinNameLength)
151:
152:     // If present, the Callsign must be 3 to 10 Callsign characters (A..Z, 0..9 and /)
153:     if (Length(edtCallsign.Text) > 0) then
154:     begin
155:         if (Length(edtCallsign.Text) < cbytMinCallsignLength) then
156:         begin
157:             ErrorMessageDlgOk('Invalid Callsign', cstrCallsignEntry +
158:                                     #13 +
159:                                     #13 +
160:                                     cstrValidCallsign);
161:             ModalResult := mrNone;
162:             edtCallsign.SetFocus;
163:             Exit;
164:         end; // if (Length(edtCallsign.Text) < cbytMinCallsignLength)
165:     end; // if (Length(edtCallsign.Text) > 0)
166:
167:     // The Registration Key must be 8 digits (0..9)
168:     if Length(edtRegKey.Text) < 8 then
169:     begin
170:         ErrorMessageDlgOk('Invalid Registration Key', cstrRegKeyEntry +
171:                             #13 +
172:                             #13 +
173:                             cstrValidRegKey);
174:         ModalResult := mrNone;
175:         edtRegKey.SetFocus;
176:         Exit;
177:     end; // if Length(edtRegKey.Text) = 0
178:
179:     // Now we check the Registration key entered against the value calculated from
180:     // the Callsign or Name. If the calculated value is the same as the entered value

```

```

181: // all is good and we consider the application "Registered", set the Registration
182: // variables and exit the dialogue.
183: //
184: // If they are not the same, we display an error message and consider the
185: // application "Not Registered", clear the Registration variables, display an error
186: // message and remain in the dialogue.
187:
188: if Length(edtCallsign.Text) > 0 then
189:     vstrRegString := edtCallsign.Text[1] + edtCallsign.Text[Length(edtCallsign.Text)]
190: else
191:     vstrRegString := edtName.Text[1] + edtName.Text[Length(edtName.Text)];
192:
193: if CalculateRegistrationKey(vstrRegString) = edtRegKey.Text then
194: begin
195:     // Registration was Successful
196:     gvblnRegistered := True;
197:     gvstrRegisteredName := edtName.Text;
198:     gvstrRegisteredCall := edtCallsign.Text;
199:     gvstrRegistrationKey := edtRegKey.Text;
200:     frmMain.DisplayTitleBar;
201: end
202: else
203: begin
204:     // OOooppss
205:     ErrorMessageDlgOk('Incorrect Registration Key', cstrRegKeyEntry +
206:                        #13 +
207:                        #13 +
208:                        cstrUnmatchedRegKey);
209:     ModalResult := mrNone;
210:     edtRegKey.SetFocus;
211: end; // if CalculateRegistrationKey('aa') = edtRegKey.Text
212:
213: end; // procedure TdlgRegister.bbtOkClick
214:
215: //=====
216: procedure TdlgRegister.bbtCancelClick(Sender: TObject);
217: begin
218:     if ConfirmationMessageDlg('Confirm Cancellation',
219:                               'Confirm you wish to cancel the Registration.') = mrNo then
220:     begin
221:         ModalResult := mrNone;
222:         edtName.SetFocus;
223:     end; // if ConfirmationMessageDlg
224: end; // procedure TdlgRegister.bbtCancelClick
225:
226: //=====
227: //                KEYPRESS ROUTINES
228: //=====
229: procedure TdlgRegister.edtNameKeyPress(Sender: TObject; var Key: char);
230: begin
231:     Key := ValidAlphaCharacter(Key); //showmessage(IntToStr(Ord(Key)));
232: end; // procedure TdlgRegister.Edit2KeyPress
233:
234: //=====
235: procedure TdlgRegister.edtCallsignKeyPress(Sender: TObject; var Key: char);
236: begin
237:     Key := ValidCallsignCharacter(Key);
238: end; // procedure TdlgRegister.edtCallsignKeyPress
239:
240: //=====

```

```
241: procedure TdlgRegister.edtRegKeyKeyPress(Sender: TObject; var Key: char);
242: begin
243:     Key := ValidDigitCharacter(Key);
244: end; // procedure TdlgRegister.edtRegKeyKeyPress
245:
246: //=====
247: end. // unit Register;
248:
```