```
 1: unit TMVFiles_FAV;
 2:
 3: {$mode objfpc}{$H+}
 4:
 5: //=============================================================================================
 6: //
 7: //  TMVFiles_FAV.pas
 8: //
 9: //  Calls: AppConstants
10: //         AppVariables
11: //         Utilities : GetToneFrequencyFromToneNr
12: //
13: //  Called By: TMVFiles : OpenTMVFile
14: //
15: //  Ver: 1.0.0
16: //
17: //  Date: 9 Aug 2013
18: //
19: //=============================================================================================
20:
21: interface
22:
23: uses
24:   Classes, SysUtils,
25:   // Application Units
26:   AppConstants, AppVariables, Utilities;
27:
28: function MakeFAVRecord(vbytRecord : Byte) : string;
29: procedure ParseFAVRecord(vbytRecNr : Byte; vstrRecord : string);
30:
31: implementation
32:
33: //=============================================================================================
34: function MakeFAVRecord(vbytRecord : Byte) : string;
35:
36: var
37:   vstrTRecord : string;
38:   vstrTOffset : string;
39:
40: begin
41:
42:   // Record Nr
43:   vstrTRecord := IntToStr(vbytRecord) + ',';
44:
45:   // VFO
46:   if gvstrFAVChannelDataArray[vbytRecord, gcbytVFOField] = gcstrVHF then
47:     vstrTRecord := vstrTRecord + gcstrTMV7VFO_VHF + ','
48:   else if gvstrFAVChannelDataArray[vbytRecord, gcbytVFOField] = gcstrUHF then
49:     vstrTRecord := vstrTRecord + gcstrTMV7VFO_UHF + ','
50:   else
51:     vstrTRecord := vstrTRecord + '' + ',';
52:
53:   // RX Frequency
54:   if Length(gvstrFAVChannelDataArray[vbytRecord, gcbytRXFrequencyField]) > 0 then
55:     vstrTRecord := vstrTRecord +
56:                    Copy(gvstrFAVChannelDataArray[vbytRecord, gcbytRXFrequencyField], 3, 3) +
57:                    '.' +
58:                    Copy(gvstrFAVChannelDataArray[vbytRecord, gcbytRXFrequencyField], 6, 3) +
59:                    ','
```

```
  else
    vstrTRecord := vstrTRecord + '' + ',';

  // Step
  If Length(gvstrFAVChannelDataArray[vbytRecord, gcbytStepField]) = 0 then
    vstrTRecord := vstrTRecord + '' + ','
  else
    vstrTRecord := vstrTRecord +
                 gvstrStepArray[StrToInt(gvstrFAVChannelDataArray[vbytRecord,
                 gcbytStepField])] + ',';

  // Shift
  If Length(gvstrFAVChannelDataArray[vbytRecord, gcbytShiftField]) = 0 then
    vstrTRecord := vstrTRecord + '' + ','
  else
    case gvstrFAVChannelDataArray[vbytRecord, gcbytShiftField] of
      gcstrShiftSimplex :
             vstrTRecord := vstrTRecord + gcstrTMV7ShiftSimplex + ',';
      gcstrShiftPlus :
             vstrTRecord := vstrTRecord + gcstrTMV7ShiftPlus + ',';
      gcstrShiftMinus :
             vstrTRecord := vstrTRecord + gcstrTMV7ShiftMinus + ',';
    end;// case gvstrFAVChannelDataArray[vbytRecord, gcbytShiftField]

  // Reverse
  If Length(gvstrFAVChannelDataArray[vbytRecord, gcbytReverseField]) = 0 then
    vstrTRecord := vstrTRecord + '' + ','
  else
    if gvstrFAVChannelDataArray[vbytRecord, gcbytReverseField] = gcstrOff then
       vstrTRecord := vstrTRecord + gcstrTMV7Off + ','
    else
      vstrTRecord := vstrTRecord + gcstrTMV7On + ',';

  //  TONE - CTCSS
  // This is sort of comlicated. We have three radio boxes which give us the correct
  // status of the TMV7 Tone functions as well as a list of tones in the combo box
  //
  // First we determine the Tone Function Status
  If (Length(gvstrFAVChannelDataArray[vbytRecord, gcbytToneField]) = 0) and
     (Length(gvstrFAVChannelDataArray[vbytRecord, gcbytCTCSSField]) = 0)
  then
  begin
    // There is no Tone Function selected so we null out the Status Field
    vstrTRecord := vstrTRecord + '' + ',';
    // and the Tone Frequency field
    vstrTRecord := vstrTRecord + '' + ',';
  end
  else
  begin
    // We have a Tone Function selected so we determine both the Function as well
    // as the Tone Freq
    if  ((gvstrFAVChannelDataArray[vbytRecord, gcbytToneField]) = gcstrOff) and
        ((gvstrFAVChannelDataArray[vbytRecord, gcbytCTCSSField]) = gcstrOff) then
    begin
      // Both Tone Functions are turned off
      vstrTRecord := vstrTRecord + gcstrTMV7None + ',';
      vstrTRecord := vstrTRecord + '' + ',';
    end
    else if ((gvstrFAVChannelDataArray[vbytRecord, gcbytToneField]) = gcstrOn) then
```

```
119:     begin
120:       // The Tone Function is turned on
121:       vstrTRecord := vstrTRecord + gcstrTMV7Tone + ',';
122:       vstrTRecord := vstrTRecord +
123:            GetToneFrequencyFromToneNr(StrToInt(gvstrFAVChannelDataArray[vbytRecord,
124:            gcbytToneNrField]))+ ',';
125:     end
126:     else
127:     begin
128:       // The CTCSS Function is tuirned on
129:       vstrTRecord := vstrTRecord + gcstrTMV7CTCSS + ',';
130:       vstrTRecord := vstrTRecord +
131:            GetToneFrequencyFromToneNr(StrToInt(gvstrFAVChannelDataArray[vbytRecord,
132:            gcbytCTCSSNrField])) + ',';
133:     end;// if  ((gvstrFAVChannelDataArray[vbytRecord, gcbytToneField])
134:
135:   end;// If (Length(gvstrFAVChannelDataArray[vbytRecord, gcbytToneField]) = 0)
136:
137:   // DTSS Function and Code
138:   If Length(gvstrFAVChannelDataArray[vbytRecord, gcbytDTSSField]) = 0 then
139:     vstrTRecord := vstrTRecord + '' + ',' + '' + ','
140:   else
141:     if gvstrFAVChannelDataArray[vbytRecord, gcbytDTSSField] = gcstrOff then
142:     begin
143:         vstrTRecord := vstrTRecord + gcstrTMV7Off + ',';
144:         vstrTRecord := vstrTRecord + '' + ',';
145:     end
146:     else
147:     begin
148:       vstrTRecord := vstrTRecord + gcstrTMV7On + ',';
149:       vstrTRecord := vstrTRecord + gvstrFAVChannelDataArray[vbytRecord,
150:       gcbytDTSSCodeField] + ','
151:     end;// if gvstrFAVChannelDataArray[vbytRecord, gcbytDTSSField]
152:
153:   // Shift Offset
154:   If Length(gvstrFAVChannelDataArray[vbytRecord, gcbytShiftOffsetField]) = 0 then
155:     vstrTRecord := vstrTRecord + '' + ','
156:   else
157:     vstrTRecord := vstrTRecord +
158:     Copy(gvstrFAVChannelDataArray[vbytRecord, gcbytShiftOffsetField], 2, 2) +
159:     '.' +
160:     Copy(gvstrFAVChannelDataArray[vbytRecord, gcbytShiftOffsetField], 4, 2) + ',';
161:
162:   // Scan
163:   If Length(gvstrFAVChannelDataArray[vbytRecord, gcbytScanField]) = 0 then
164:     vstrTRecord := vstrTRecord + '' + ','
165:   else
166:     if gvstrFAVChannelDataArray[vbytRecord, gcbytScanField] = gcstrOff then
167:         vstrTRecord := vstrTRecord + gcstrTMV7Off + ','
168:     else
169:       vstrTRecord := vstrTRecord + gcstrTMV7On + ',';
170:
171:   // RF Power
172:   If Length(gvstrFAVChannelDataArray[vbytRecord, gcbytRFPowerField]) = 0 then
173:     vstrTRecord := vstrTRecord + '' + ','
174:   else
175:     case gvstrFAVChannelDataArray[vbytRecord, gcbytRFPowerField] of
176:       gcstrRFPowerLow :
177:         vstrTRecord := vstrTRecord + gcstrTMV7RFPowerLow + ',';
```

```
178:       gcstrRFPowerMedium :
179:         vstrTRecord := vstrTRecord + gcstrTMV7RFPowerMedium + ',';
180:       gcstrRFPowerHigh :
181:         vstrTRecord := vstrTRecord + gcstrTMV7RFPowerHigh + ',';
182:     end;// case gvstrFAVChannelDataArray[vbytRecord, gcbytRFPowerField]
183:
184:   // Button Name
185:   If Length(gvstrFAVChannelDataArray[vbytRecord, gcbytChannelNameField]) = 0 then
186:     vstrTRecord := vstrTRecord + '' + ','
187:   else
188:     vstrTRecord := vstrTRecord + gvstrFAVChannelDataArray[vbytRecord,
189:                    gcbytChannelNameField] + ',';
190:
191:   // Comments
192:   If Length(gvstrFAVChannelDataArray[vbytRecord, gcbytCommentsField]) = 0 then
193:     vstrTRecord := vstrTRecord + ''
194:   else
195:     vstrTRecord := vstrTRecord + gvstrFAVChannelDataArray[vbytRecord, gcbytCommentsField];
196:
197:   Result := vstrTRecord;
198:
199: end;// function MakeFAVRecord
200:
201: //============================================================================================
202: procedure ParseFAVRecord(vbytRecNr : Byte; vstrRecord : string);
203:
204: var
205:   vbytCommaPos : Byte;
206:   vstrTStr : string;
207:   vbytTbyt : Byte;
208:   vstrTToneNr : string;
209:
210: begin
211:
212:   // Bypass the Record Nr
213:   vbytCommaPos := Pos(',', vstrRecord );
214:   vstrRecord := Copy(vstrRecord, vbytCommaPos+1, Length(vstrRecord));
215:
216:   // VFO
217:   vbytCommaPos := Pos(',', vstrRecord );
218:   vstrTStr := Copy(vstrRecord, 1, vbytCommaPos-1);
219:   case  vstrTstr of
220:     '' :     gvstrFAVChannelDataArray[vbytRecNr, gcbytVFOField] := '';
221:     gcstrTMV7VFO_VHF : gvstrFAVChannelDataArray[vbytRecNr, gcbytVFOField] := gcstrVHFVFO;
222:     else
223:       gvstrFAVChannelDataArray[vbytRecNr, gcbytVFOField] :=gcstrUHFVFO;
224:   end;// case of vstrTstr
225:   vstrRecord := Copy(vstrRecord, vbytCommaPos+1, Length(vstrRecord));
226:
227:   // RX Frequency
228:   vbytCommaPos := Pos(',', vstrRecord );
229:   vstrTStr := Copy(vstrRecord, 1, vbytCommaPos-1);
230:   if Length(vstrTStr) > 0 then
231:     vstrTStr := '00' + Copy(vstrRecord, 1, 3) + Copy(vstrRecord, 5, 3) + '000';
232:   gvstrFAVChannelDataArray[vbytRecNr, gcbytRXFrequencyField] := vstrTStr;
233:   vstrRecord := Copy(vstrRecord, vbytCommaPos+1, Length(vstrRecord));
234:
235:   // Step Size
236:   vbytCommaPos := Pos(',', vstrRecord );
```

```
237:   vstrTStr := Copy(vstrRecord, 1, vbytCommaPos-1);
238:   if Length(vstrTStr) > 0 then
239:   begin
240:     // We look for a decimal to determine if we have to conver to Real
241:     if Pos('.', vstrTStr) > 0 then
242:       vbytTByt := GetStepIndex(StrToFloat(vstrTStr))
243:     else
244:       vbytTByt := GetStepIndex(StrToFloat(vstrTStr + '.0'));
245:     gvstrFAVChannelDataArray[vbytRecNr, gcbytStepField] := IntToStr(vbytTByt);
246:   end
247:   else
248:     gvstrFAVChannelDataArray[vbytRecNr, gcbytStepField] := '';
249:   vstrRecord := Copy(vstrRecord, vbytCommaPos+1, Length(vstrRecord));
250:
251:   // Shift
252:   vbytCommaPos := Pos(',', vstrRecord );
253:   vstrTStr := Copy(vstrRecord, 1, vbytCommaPos-1);
254:   case vstrTStr of
255:     '' : gvstrFAVChannelDataArray[vbytRecNr, gcbytShiftField] := '';
256:     gcstrTMV7ShiftSimplex :
257:       gvstrFAVChannelDataArray[vbytRecNr, gcbytShiftField] := gcstrShiftSimplex;
258:     gcstrTMV7ShiftPlus :
259:       gvstrFAVChannelDataArray[vbytRecNr, gcbytShiftField] := gcstrShiftPlus;
260:     gcstrTMV7ShiftMinus :
261:       gvstrFAVChannelDataArray[vbytRecNr, gcbytShiftField] := gcstrShiftMinus;
262:   end;// case vstrTStr of
263:   vstrRecord := Copy(vstrRecord, vbytCommaPos+1, Length(vstrRecord));
264:
265:   // Reverse
266:   vbytCommaPos := Pos(',', vstrRecord );
267:   vstrTStr := Copy(vstrRecord, 1, vbytCommaPos-1);
268:   case vstrTStr of
269:     '' : gvstrFAVChannelDataArray[vbytRecNr, gcbytReverseField] := '';
270:     gcstrTMV7Off : gvstrFAVChannelDataArray[vbytRecNr, gcbytReverseField] := gcstrOff;
271:     else
272:       gvstrFAVChannelDataArray[vbytRecNr, gcbytReverseField] := gcstrOn;
273:   end;// case vstrTStr of
274:   vstrRecord := Copy(vstrRecord, vbytCommaPos+1, Length(vstrRecord));
275:
276:   // Tone Function - This takes care of both Tone and CTCSS On/Off fields
277:   vbytCommaPos := Pos(',', vstrRecord );
278:   vstrTStr := Copy(vstrRecord, 1, vbytCommaPos-1);
279:   case vstrTStr of
280:     '' : begin
281:            gvstrFAVChannelDataArray[vbytRecNr, gcbytToneField] := '';
282:            gvstrFAVChannelDataArray[vbytRecNr, gcbytCTCSSField] := '';
283:          end;// '', cstrNone
284:     gcstrTMV7None : begin
285:            gvstrFAVChannelDataArray[vbytRecNr, gcbytToneField] := gcstrOff;
286:            gvstrFAVChannelDataArray[vbytRecNr, gcbytCTCSSField] := gcstrOff;
287:          end;// '', cstrNone
288:     gcstrTMV7Tone : begin
289:            gvstrFAVChannelDataArray[vbytRecNr, gcbytToneField] := gcstrOn;
290:            gvstrFAVChannelDataArray[vbytRecNr, gcbytCTCSSField] := gcstrOff;
291:          end;// cstrTone
292:     else
293:       begin
294:         gvstrFAVChannelDataArray[vbytRecNr, gcbytToneField] := gcstrOff;
295:         gvstrFAVChannelDataArray[vbytRecNr, gcbytCTCSSField] := gcstrOn;
```

```
296:       end;
297:   end;// case vstrTStr of
298:   vstrRecord := Copy(vstrRecord, vbytCommaPos+1, Length(vstrRecord));
299:
300:   // Tone Frequency - This takes care of both Tone and CTCSS Frequency fields
301:   vbytCommaPos := Pos(',', vstrRecord );
302:   vstrTStr := Copy(vstrRecord, 1, vbytCommaPos-1);
303:   if Length(vstrTStr) > 0 then
304:   begin
305:     // There is a Frequency in the record. That means that either Tone or CTCSS have
306:     // been selected. vstrTStr contains the Tone Frequency as a string
307:     if gvstrFAVChannelDataArray[vbytRecNr, gcbytToneField] = gcstrOn then
308:     begin
309:       // Tone has been selected so we have to populate the Tone Nr field and Default the
310:       // CTCSS Nr field
311:       vbytTByt := GetToneNrFromFrequency(vstrTStr);
312:       if vbytTByt < 10 then
313:         vstrTToneNr := '0' + IntToStr(vbytTByt)
314:       else
315:         vstrTToneNr := IntToStr(vbytTByt);
316:       if Length(vstrTToneNr) = 1 then
317:         vstrTToneNr := '0' + vstrTToneNr;
318:       gvstrFAVChannelDataArray[vbytRecNr, gcbytToneNrField] := vstrTToneNr;
319:       gvstrFAVChannelDataArray[vbytRecNr, gcbytCTCSSNrField] := '01';
320:
321:     end
322:     else
323:     begin
324:       // CTCSS has been selected so we have to populate the CTCSS Nr field and Default the
325:       // Tone Nr field
326:       vbytTByt := GetToneNrFromFrequency(vstrTStr);
327:       vstrTToneNr := IntToStr(vbytTByt);
328:       if Length(vstrTToneNr) = 1 then
329:         vstrTToneNr := '0' + vstrTToneNr;
330:       gvstrFAVChannelDataArray[vbytRecNr, gcbytCTCSSNrField] := vstrTToneNr;
331:       gvstrFAVChannelDataArray[vbytRecNr, gcbytToneNrField] := '01';
332:     end;//éé if gvstrFAVChannelDataArray[vbytRecNr, gcbytToneField] = gcstrOn
333:
334:   end
335:   else
336:   begin
337:     // There is no Tone Frequency in the record so we clear the Tone Nr fields
338:     gvstrFAVChannelDataArray[vbytRecNr, gcbytToneNrField] := '01';
339:     gvstrFAVChannelDataArray[vbytRecNr, gcbytCTCSSNrField] := '01';
340:   end;//  if Length(vstrTStr) > 0
341:   vstrRecord := Copy(vstrRecord, vbytCommaPos+1, Length(vstrRecord));
342:
343:   // DTSS On/Off
344:   vbytCommaPos := Pos(',', vstrRecord );
345:   vstrTStr := Copy(vstrRecord, 1, vbytCommaPos-1);
346:   case vstrTStr of
347:     '' : gvstrFAVChannelDataArray[vbytRecNr, gcbytDTSSField] := '';
348:     gcstrTMV7Off : gvstrFAVChannelDataArray[vbytRecNr, gcbytDTSSField] := gcstrOff;
349:     else
350:       gvstrFAVChannelDataArray[vbytRecNr, gcbytDTSSField] := gcstrOn;
351:   end;// case vstrTStr of
352:   vstrRecord := Copy(vstrRecord, vbytCommaPos+1, Length(vstrRecord));
353:
354:   // DTSS Code
```

```
355:   vbytCommaPos := Pos(',', vstrRecord );
356:   vstrTStr := Copy(vstrRecord, 1, vbytCommaPos-1);
357:   case vstrTStr of
358:     '' : gvstrFAVChannelDataArray[vbytRecNr, gcbytDTSSCodeField] := '000';
359:     else
360:       gvstrFAVChannelDataArray[vbytRecNr, gcbytDTSSCodeField] := vstrTStr;
361:   end;// case vstrTStr of
362:   vstrRecord := Copy(vstrRecord, vbytCommaPos+1, Length(vstrRecord));
363:
364:     // Shift Offset
365:   vbytCommaPos := Pos(',', vstrRecord );
366:   vstrTStr := Copy(vstrRecord, 1, vbytCommaPos-1);
367:   case vstrTStr of
368:     '' : gvstrFAVChannelDataArray[vbytRecNr, gcbytShiftOffsetField] := '';
369:     else
370:       gvstrFAVChannelDataArray[vbytRecNr, gcbytShiftOffsetField] := '0' +
371:           Copy(vstrTStr, 1, 2) + Copy(vstrTStr, 4, 2) + '0000';
372:   end;// case vstrTStr of
373:   vstrRecord := Copy(vstrRecord, vbytCommaPos+1, Length(vstrRecord));
374:
375:   // Scan On/Off
376:   vbytCommaPos := Pos(',', vstrRecord );
377:   vstrTStr := Copy(vstrRecord, 1, vbytCommaPos-1);
378:   case vstrTStr of
379:     '' : gvstrFAVChannelDataArray[vbytRecNr, gcbytScanField] := '';
380:     gcstrTMV7Off : gvstrFAVChannelDataArray[vbytRecNr, gcbytScanField] := gcstrOff;
381:     else
382:       gvstrFAVChannelDataArray[vbytRecNr, gcbytScanField] := gcstrOn;
383:   end;// case vstrTStr of
384:   vstrRecord := Copy(vstrRecord, vbytCommaPos+1, Length(vstrRecord));
385:
386:   // RF Power
387:   vbytCommaPos := Pos(',', vstrRecord );
388:   vstrTStr := Copy(vstrRecord, 1, vbytCommaPos-1);
389:   case vstrTStr of
390:     '' : gvstrFAVChannelDataArray[vbytRecNr, gcbytRFPowerField] := '';
391:     gcstrTMV7RFPowerLow :
392:       gvstrFAVChannelDataArray[vbytRecNr, gcbytRFPowerField] := gcstrRFPowerLow;
393:     gcstrTMV7RFPowerMedium :
394:       gvstrFAVChannelDataArray[vbytRecNr, gcbytRFPowerField] := gcstrRFPowerMedium;
395:     gcstrTMV7RFPowerHigh :
396:       gvstrFAVChannelDataArray[vbytRecNr, gcbytRFPowerField] := gcstrRFPowerHigh;
397:   end;// case vstrTStr of
398:   vstrRecord := Copy(vstrRecord, vbytCommaPos+1, Length(vstrRecord));
399:
400:   // Channel Name
401:   vbytCommaPos := Pos(',', vstrRecord );
402:   vstrTStr := Copy(vstrRecord, 1, vbytCommaPos-1);
403:   gvstrFAVChannelDataArray[vbytRecNr, gcbytChannelNameField] := vstrTStr;
404:   vstrRecord := Copy(vstrRecord, vbytCommaPos+1, Length(vstrRecord));
405:
406:   // Comments
407:   gvstrFAVChannelDataArray[vbytRecNr, gcbytCommentsField] := vstrRecord;
408:
409: end;// procedure ParseFAVRecord;
410:
411: //============================================================================
412: end.// unit TMVFiles_FAV;
413:
```