

```

1: unit TMVFileReport;
2:
3: {$mode delphi}
4:
5: //=====
6: //
7: //   TMVFileReport.pas
8: //
9: //   Calls:
10: //
11: //   Called By:
12: //
13: //   Ver: 1.0.0
14: //
15: //   Date: 24 Apr 2014
16: //
17: //=====
18:
19:
20: interface
21:
22: uses
23:   Classes, db, SysUtils, SdfData, FileUtil, LR_Class, LR_DBSet, Forms, Controls,
24:   Graphics, Dialogs, StdCtrls;
25:
26: type
27:
28:   { TfrmTMVFileReport }
29:
30:   TfrmTMVFileReport = class(TForm)
31:     edtCHName: TEdit;
32:     edtRXFrequency: TEdit;
33:     edtCHNR: TEdit;
34:     frDBDTMFDataSet: TfrDBDataSet;
35:     frDBFAVDataSet: TfrDBDataSet;
36:     frDBUHFDDataSet: TfrDBDataSet;
37:     frDBVHFDDataSet: TfrDBDataSet;
38:     frTMVFileReport: TfrReport;
39:     SdfDTMFDataSet: TSdfDataSet;
40:     SdfFAVDataSet: TSdfDataSet;
41:     SdfUHFDDataSet: TSdfDataSet;
42:     SdfVHFDDataSet: TSdfDataSet;
43:     procedure FormActivate(Sender: TObject);
44:     procedure FormCreate(Sender: TObject);
45:     procedure frVHFReportGetValue(const ParName: String; var ParValue: Variant);
46:   private
47:     { private declarations }
48:   public
49:     { public declarations }
50:     procedure CreateTmpFiles(vstrFileName : string);
51:   end;
52:
53: var
54:   frmTMVFileReport: TfrmTMVFileReport;
55:
56: implementation
57:
58: {$R *.lfm}
59:
60: const

```

```

61:   cstrDTMFFileName = 'DTMF.TMP';
62:   cstrFAVFileName = 'FAV.TMP';
63:   cstrUHFFFileName = 'UHF.TMP';
64:   cstrVHFFFileName = 'VHF.TMP';
65:
66:   cstrDelimiter = ',';
67:
68:   cstrCHNRFieldDef = 'CHNR';
69:   cstrVFOFieldDef = 'VFO';
70:   cstrRXFREQFieldDef = 'RXFREQ';
71:   cstrSTEPFieldDef = 'STEP';
72:   cstrSHIFTFieldDef = 'SHIFT';
73:   cstrREVERSEFieldDef = 'REVERSE';
74:   cstrTONEFieldDef = 'TONE';
75:   cstrTFREQFieldDef = 'TFREQ';
76:   cstrDTSSFieldDef = 'DTSS';
77:   cstrDTSSCODEFieldDef = 'DTSSCODE';
78:   cstrSHIFTOFFSETFieldDef = 'SHIFTOFFSET';
79:   cstrSCANFieldDef = 'SCAN';
80:   cstrRFPOWERFieldDef = 'RFPOWER';
81:   cstrCHNAMEFieldDef = 'CHNAME';
82:   cstrCOMMENTSFieldDef = 'COMMENTS';
83:
84:   cstrVHFUHFHeader = cstrCHNRFieldDef + cstrDelimiter +
85:                       cstrVFOFieldDef + cstrDelimiter +
86:                       cstrRXFREQFieldDef + cstrDelimiter +
87:                       cstrSTEPFieldDef + cstrDelimiter +
88:                       cstrSHIFTFieldDef + cstrDelimiter +
89:                       cstrREVERSEFieldDef + cstrDelimiter +
90:                       cstrTONEFieldDef + cstrDelimiter +
91:                       cstrTFREQFieldDef + cstrDelimiter +
92:                       cstrDTSSFieldDef + cstrDelimiter +
93:                       cstrDTSSCODEFieldDef + cstrDelimiter +
94:                       cstrSHIFTOFFSETFieldDef + cstrDelimiter +
95:                       cstrSCANFieldDef + cstrDelimiter +
96:                       cstrRFPOWERFieldDef + cstrDelimiter +
97:                       cstrCHNAMEFieldDef + cstrDelimiter +
98:                       cstrCOMMENTSFieldDef + cstrDelimiter;
99:
100:  cstrCODENRFieldDef = 'CODENR';
101:  cstrCODEFieldDef = 'CODE';
102:
103:  cstrDTMFHeader = cstrCODENRFieldDef + cstrDelimiter +
104:                  cstrCODEFieldDef + cstrDelimiter;
105:
106:
107:  var
108:    vfilTMVFile : TextFile;
109:    vstrTMVFileName : string;
110:
111:  { TfrmTMVFileReport }
112:
113:
114:  vstrTMVFilePath : string;
115:  vstrDTMFFileName : string;
116:  vfilFAVFile : TextFile;
117:  vstrFAVFileName : string;
118:  vfilDTMFFile : TextFile;
119:  vstrUHFFFileName : string;
120:  vfilUHFFFile : TextFile;

```

```

121:     vstrVHFFileName : string;
122:     vfilVHFFile : TextFile;
123:
124: //=====
125: //          FORM ROUTINES
126: //=====
127: procedure TfrmTMVFileReport.FormCreate(Sender: TObject);
128: begin
129:
130:     // Set up the VHF Dataset
131:     sdfVHFDataSet.FieldDefs.Add(cstrCHNRFieldDef, ftString);
132:     sdfVHFDataSet.Schema.Add(cstrCHNRFieldDef);
133:     sdfVHFDataSet.FieldDefs.Add(cstrVFOFieldDef, ftString);
134:     sdfVHFDataSet.Schema.Add(cstrVFOFieldDef);
135:     sdfVHFDataSet.FieldDefs.Add(cstrRXFREQFieldDef, ftString);
136:     sdfVHFDataSet.Schema.Add(cstrRXFREQFieldDef);
137:     sdfVHFDataSet.FieldDefs.Add(cstrSTEPFieldDef, ftString);
138:     sdfVHFDataSet.Schema.Add(cstrSTEPFieldDef);
139:     sdfVHFDataSet.FieldDefs.Add(cstrSHIFTFieldDef, ftString);
140:     sdfVHFDataSet.Schema.Add(cstrSHIFTFieldDef);
141:     sdfVHFDataSet.FieldDefs.Add(cstrREVERSEFieldDef, ftString);
142:     sdfVHFDataSet.Schema.Add(cstrREVERSEFieldDef);
143:     sdfVHFDataSet.FieldDefs.Add(cstrTONEFieldDef, ftString);
144:     sdfVHFDataSet.Schema.Add(cstrTONEFieldDef);
145:     sdfVHFDataSet.FieldDefs.Add(cstrTFREQFieldDef, ftString);
146:     sdfVHFDataSet.Schema.Add(cstrTFREQFieldDef);
147:     sdfVHFDataSet.FieldDefs.Add(cstrDTSSFieldDef, ftString);
148:     sdfVHFDataSet.Schema.Add(cstrDTSSFieldDef);
149:     sdfVHFDataSet.FieldDefs.Add(cstrDTSSCODEFieldDef, ftString);
150:     sdfVHFDataSet.Schema.Add(cstrDTSSCODEFieldDef);
151:     sdfVHFDataSet.FieldDefs.Add(cstrSHIFTOFFSETFieldDef, ftString);
152:     sdfVHFDataSet.Schema.Add(cstrSHIFTOFFSETFieldDef);
153:     sdfVHFDataSet.FieldDefs.Add(cstrSCANFieldDef, ftString);
154:     sdfVHFDataSet.Schema.Add(cstrSCANFieldDef);
155:     sdfVHFDataSet.FieldDefs.Add(cstrRFPOWERFieldDef, ftString);
156:     sdfVHFDataSet.Schema.Add(cstrRFPOWERFieldDef);
157:     sdfVHFDataSet.FieldDefs.Add(cstrCHNAMEFieldDef, ftString);
158:     sdfVHFDataSet.Schema.Add(cstrCHNAMEFieldDef);
159:     sdfVHFDataSet.FieldDefs.Add(cstrCOMMENTSFieldDef, ftString);
160:     sdfVHFDataSet.Schema.Add(cstrCOMMENTSFieldDef);
161:
162:     // Set up the UHF Dataset
163:     sdfUHFDataset.FieldDefs.Add(cstrCHNRFieldDef, ftString);
164:     sdfUHFDataset.Schema.Add(cstrCHNRFieldDef);
165:     sdfUHFDataset.FieldDefs.Add(cstrVFOFieldDef, ftString);
166:     sdfUHFDataset.Schema.Add(cstrVFOFieldDef);
167:     sdfUHFDataset.FieldDefs.Add(cstrRXFREQFieldDef, ftString);
168:     sdfUHFDataset.Schema.Add(cstrRXFREQFieldDef);
169:     sdfUHFDataset.FieldDefs.Add(cstrSTEPFieldDef, ftString);
170:     sdfUHFDataset.Schema.Add(cstrSTEPFieldDef);
171:     sdfUHFDataset.FieldDefs.Add(cstrSHIFTFieldDef, ftString);
172:     sdfUHFDataset.Schema.Add(cstrSHIFTFieldDef);
173:     sdfUHFDataset.FieldDefs.Add(cstrREVERSEFieldDef, ftString);
174:     sdfUHFDataset.Schema.Add(cstrREVERSEFieldDef);
175:     sdfUHFDataset.FieldDefs.Add(cstrTONEFieldDef, ftString);
176:     sdfUHFDataset.Schema.Add(cstrTONEFieldDef);
177:     sdfUHFDataset.FieldDefs.Add(cstrTFREQFieldDef, ftString);
178:     sdfUHFDataset.Schema.Add(cstrTFREQFieldDef);
179:     sdfUHFDataset.FieldDefs.Add(cstrDTSSFieldDef, ftString);
180:     sdfUHFDataset.Schema.Add(cstrDTSSFieldDef);

```

```
181:   sdfUHFDDataSet.FieldDefs.Add(cstrDTSSCODEFieldDef, ftString);
182:   sdfUHFDDataSet.Schema.Add(cstrDTSSCODEFieldDef);
183:   sdfUHFDDataSet.FieldDefs.Add(cstrSHIFTOFFSETFieldDef, ftString);
184:   sdfUHFDDataSet.Schema.Add(cstrSHIFTOFFSETFieldDef);
185:   sdfUHFDDataSet.FieldDefs.Add(cstrSCANFieldDef, ftString);
186:   sdfUHFDDataSet.Schema.Add(cstrSCANFieldDef);
187:   sdfUHFDDataSet.FieldDefs.Add(cstrRFPOWERFieldDef, ftString);
188:   sdfUHFDDataSet.Schema.Add(cstrRFPOWERFieldDef);
189:   sdfUHFDDataSet.FieldDefs.Add(cstrCHNAMEFieldDef, ftString);
190:   sdfUHFDDataSet.Schema.Add(cstrCHNAMEFieldDef);
191:   sdfUHFDDataSet.FieldDefs.Add(cstrCOMMENTSFieldDef, ftString);
192:   sdfUHFDDataSet.Schema.Add(cstrCOMMENTSFieldDef);
193:
194:   // Set up the FAV Dataset
195:   sdfFAVDataSet.FieldDefs.Add(cstrCHNRFieldDef, ftString);
196:   sdfFAVDataSet.Schema.Add(cstrCHNRFieldDef);
197:   sdfFAVDataSet.FieldDefs.Add(cstrVFOFieldDef, ftString);
198:   sdfFAVDataSet.Schema.Add(cstrVFOFieldDef);
199:   sdfFAVDataSet.FieldDefs.Add(cstrRXFREQFieldDef, ftString);
200:   sdfFAVDataSet.Schema.Add(cstrRXFREQFieldDef);
201:   sdfFAVDataSet.FieldDefs.Add(cstrSTEPFieldDef, ftString);
202:   sdfFAVDataSet.Schema.Add(cstrSTEPFieldDef);
203:   sdfFAVDataSet.FieldDefs.Add(cstrSHIFTFieldDef, ftString);
204:   sdfFAVDataSet.Schema.Add(cstrSHIFTFieldDef);
205:   sdfFAVDataSet.FieldDefs.Add(cstrREVERSEFieldDef, ftString);
206:   sdfFAVDataSet.Schema.Add(cstrREVERSEFieldDef);
207:   sdfFAVDataSet.FieldDefs.Add(cstrTONEFieldDef, ftString);
208:   sdfFAVDataSet.Schema.Add(cstrTONEFieldDef);
209:   sdfFAVDataSet.FieldDefs.Add(cstrTFREQFieldDef, ftString);
210:   sdfFAVDataSet.Schema.Add(cstrTFREQFieldDef);
211:   sdfFAVDataSet.FieldDefs.Add(cstrDTSSFieldDef, ftString);
212:   sdfFAVDataSet.Schema.Add(cstrDTSSFieldDef);
213:   sdfFAVDataSet.FieldDefs.Add(cstrDTSSCODEFieldDef, ftString);
214:   sdfFAVDataSet.Schema.Add(cstrDTSSCODEFieldDef);
215:   sdfFAVDataSet.FieldDefs.Add(cstrSHIFTOFFSETFieldDef, ftString);
216:   sdfFAVDataSet.Schema.Add(cstrSHIFTOFFSETFieldDef);
217:   sdfFAVDataSet.FieldDefs.Add(cstrSCANFieldDef, ftString);
218:   sdfFAVDataSet.Schema.Add(cstrSCANFieldDef);
219:   sdfFAVDataSet.FieldDefs.Add(cstrRFPOWERFieldDef, ftString);
220:   sdfFAVDataSet.Schema.Add(cstrRFPOWERFieldDef);
221:   sdfFAVDataSet.FieldDefs.Add(cstrCHNAMEFieldDef, ftString);
222:   sdfFAVDataSet.Schema.Add(cstrCHNAMEFieldDef);
223:   sdfFAVDataSet.FieldDefs.Add(cstrCOMMENTSFieldDef, ftString);
224:   sdfFAVDataSet.Schema.Add(cstrCOMMENTSFieldDef);
225:
226:   // Set up the DTMF Dataset
227:   sdfDTMFDataSet.FieldDefs.Add(cstrCODENRFieldDef, ftString);
228:   sdfDTMFDataSet.Schema.Add(cstrCODENRFieldDef);
229:   sdfDTMFDataSet.FieldDefs.Add(cstrCODEFieldDef, ftString);
230:   sdfDTMFDataSet.Schema.Add(cstrCODEFieldDef);
231:
232: end; // procedure TfrmTMVFileReport.FormCreate
233:
234: procedure TfrmTMVFileReport.FormActivate(Sender: TObject);
235: begin
236:   edtCHNR.text := sdfVHFDDataSet.FieldValues[cstrCHNRFieldDef];
237:   edtRXFREQUENCY.text := sdfVHFDDataSet.FieldValues[cstrRXFREQFieldDef];
238:   edtCHNAME.text := sdfVHFDDataSet.FieldValues[cstrCHNAMEFieldDef];
239: end; // procedure TfrmTMVFileReport.FormActivate
240:
```

```
241: procedure TfrmTMVFileReport.CreateTmpFiles(vstrFileName : string);
242: var
243:   vbytTemp : byte;
244:   vstrTStr : string;
245:
246: begin
247:
248:   vstrTMVFileName := vstrFileName;
249:
250:   // Get the TMV Filepath so we can create the TMP files in the same folder
251:   vstrTMVFilePath := ExtractFileDir(vstrTMVFileName);
252:
253:   // Open the file for Reading
254:   AssignFile(vfilTMVFile, vstrTMVFileName);
255:   Reset(vfilTMVFile);
256:
257:   // Read the first line to validate the file. If it is [TMVFile Ver. 1.0.0] it is
258:   // a valid file
259:   Readln(vfilTMVFile, vstrTStr);
260:
261:   if vstrTStr <> '[TMVFile Ver. 1.0.0]' then
262:   begin
263:     ShowMessage('Invalid TMVFile');
264:     CloseFile(vfilTMVFile);
265:     Exit;
266:   end; // if vsrtTStr <> '[TMVFile Ver. 1.0.0]'
267:
268:
269:
270:   //=====
271:   // We have a valid input file so now we create the TMP VHF file
272:   //=====
273:   vstrVHFFFileName := vstrTMVFilePath + '\' + cstrVHFFFileName;
274:   AssignFile(vfilVHFFFile, vstrVHFFFileName);
275:   Rewrite(vfilVHFFFile);
276:
277:   // Now bypass the VHF header
278:   ReadLn(vfilTMVFile, vstrTStr);
279:
280:   // Create the VHF-UHF CSV Header and read all of the VHF data into the TMP file
281:   Writeln(vfilVHFFFile, cstrVHFUHFHeader);
282:
283:   for vbytTemp := 1 to 99 do
284:   begin
285:     ReadLn(vfilTMVFile, vstrTStr);
286:     Writeln(vfilVHFFFile, vstrTStr);
287:   end;
288:
289:   // Close the VHF file
290:   CloseFile(vfilVHFFFile);
291:
292:   // and Open the VHF Database
293:   SdfVHFDataSet.FileName := vstrVHFFFileName;
294:   sdfVHFDataSet.Active := True;
295:   sdfVHFDataSet.First;
296:
297:   //=====
298:   // Now we create the TMP UHF file
299:   //=====
300:   vstrUHFFFileName := vstrTMVFilePath + '\' + cstrUHFFFileName;
```

```
301: AssignFile(vfilUHFFile, vstrUHFFFileName);
302: Rewrite(vfilUHFFile);
303:
304: // and Reset the TMV File and bypassthe UHF header
305: Reset(vfilTMVFile);
306: repeat
307:     ReadLn(vfilTMVFile, vstrTStr);
308: until vstrTStr = '[UHF MEMORY]';
309:
310: // Create the VHF-UHF CSV Header and read all of the UHF data into the TMP file
311: Writeln(vfilUHFFile, cstrVHFUHFHeader);
312:
313: for vbytTemp := 1 to 99 do
314: begin
315:     ReadLn(vfilTMVFile, vstrTStr);
316:     Writeln(vfilUHFFile, vstrTStr);
317: end;
318:
319: // Close both files
320: CloseFile(vfilTMVFile);
321: CloseFile(vfilUHFFile);
322:
323: // and Open the UHF Database
324: SdfUHFDDataSet.FileName := vstrUHFFFileName;
325: SdfUHFDDataSet.Active := True;
326: SdfUHFDDataSet.First;
327:
328: //=====
329: // Now we create the TMP FAV file
330: //=====
331: vstrFAVFileName := vstrTMVFilePath + '\' + cstrFAVFileName;
332: AssignFile(vfilFAVFile, vstrFAVFileName);
333: Rewrite(vfilFAVFile);
334:
335: // and Reset the TMV File and bypassthe FAV header
336: Reset(vfilTMVFile);
337: repeat
338:     ReadLn(vfilTMVFile, vstrTStr);
339: until vstrTStr = '[FAV MEMORY]';
340:
341: // Create the VHF-UHF CSV Header and read all of the UHF data into the TMP file
342: Writeln(vfilFAVFile, cstrVHFUHFHeader);
343:
344: for vbytTemp := 1 to 12 do
345: begin
346:     ReadLn(vfilTMVFile, vstrTStr);
347:     Writeln(vfilFAVFile, vstrTStr);
348: end;
349:
350: // Close both files
351: CloseFile(vfilTMVFile);
352: CloseFile(vfilFAVFile);
353:
354: // and Open the FAV Database
355: sdfFAVDataSet.FileName := vstrFAVFileName;
356: SdfFAVDataSet.Active := True;
357: SdfFAVDataSet.First;
358:
359:
360:
```

```

361:
362:
363:
364: //=====
365: // Now we create the TMP DTMF file
366: //=====
367: vstrDTMFFFileName := vstrTMVFilePath + '\' + cstrDTMFFFileName;
368: AssignFile(vfilDTMFFFile, vstrDTMFFFileName);
369: Rewrite(vfilDTMFFFile);
370:
371: // and Reset the TMV File and bypassthe FAV header
372: Reset(vfilTMVFile);
373: repeat
374:   ReadLn(vfilTMVFile, vstrTStr);
375: until vstrTStr = '[DTMF MEMORY]';
376:
377: // Create the DTMF CSV Header and read all of the UHF data into the TMP file
378: Writeln(vfilDTMFFFile, cstrDTMFHeader);
379:
380: for vbytTemp := 1 to 10 do
381: begin
382:   ReadLn(vfilTMVFile, vstrTStr);
383:   Writeln(vfilDTMFFFile, vstrTStr);
384: end;
385:
386: // Close both files
387: CloseFile(vfilTMVFile);
388: CloseFile(vfilDTMFFFile);
389:
390: // and Open the FAV Database
391: sdfDTMFDataSet.FileName := vstrDTMFFFileName;
392: sdfDTMFDataSet.Active := True;
393: sdfDTMFDataSet.First;
394:
395:
396:
397:
398:
399:
400:
401:
402: end;
403:
404: procedure TfrmTMVFileReport.frVHFReportGetValue(const ParName: String; var ParValue: Variant);
405: begin
406:   if ParName = 'rpvTMVFileName' then
407:     ParValue := ExtractFileName(vstrTMVFileName);
408:   if ParName = 'rvtDTMFHeader' then
409:     ParValue := cstrDTMFHeader;
410: end; // procedure TfrmTMVFileReport.frVHFReportGetValue
411:
412: //=====
413: end. // unit TMVFileReport;
414:

```