1. EQ DATA

procedure EQData(sgData : TStringGrid);

var

i, j, iNumS :integer;

xfloat : double;

xChannel, xStaName, s, x, y, xLat, xLon : string;

begin

// frmMain.sgResult.RowCount := frmMain.sgAtlasPick.RowCount;

// sgData.RowCount := frmMain.sgAtlasPick.RowCount;

x := EQPlotter\_data.CreatePlotDatFirstLine(frmMain.sgAtlasData\_Final.Cells[4,1]);

// frmMain.ledFileName.Text := frmMain.sgAtlasPick.Cells[4,4];

xStaName := frmMain.sgAtlasData\_Final.Cells[1,1]; // first Station

xStaName := LeftStr(xStaName, 3);

s := x +' '+ Utils\_DB.FirstStationLatLon(xStaName, frmMain.lblProgramPath.Caption + 'DB\'); // extract First Station LatLon

// frmMain.mmoPlot\_dat.Lines.Add(s);

sgData.Cells[0,0] := s; // write PLOT.DATA heading: 0214190237 47.5 08.15 124.03 000.0

iNumS := 0;

for i := 1 to frmMain.sgAtlasData\_Final.RowCount-1 do begin

if Trim(frmMain.sgAtlasData\_Final.Cells[1, i]) = '' then break;

//----- station

x := LeftStr(frmMain.sgAtlasData\_Final.Cells[1, i], 3);

s := x +' ';

sgData.Cells[0,i] := x;

//----- P minute + second

x := Utils.ExtractDateTime(frmMain.sgAtlasData\_Final.Cells[4, i]);

s := s + x;

sgData.Cells[1,i] := x;

//----- S minute + second

if Trim(frmMain.sgAtlasData\_Final.Cells[6, i]) <> ''

then begin

x := Utils.ExtractDateTime(frmMain.sgAtlasData\_Final.Cells[6, i]);

s := s + x;

sgData.Cells[4,i] := x;

inc(iNumS);

end

else s := s + ' ';

//----- CODA

if Trim(frmMain.sgAtlasData\_Final.Cells[12, i]) <> '' then begin

x := frmMain.sgAtlasData\_Final.Cells[12, i];

s := s +' '+ x;

sgData.Cells[7,i] := x;

end;

end;

//------ Set the final rowcount

sgData.RowCount := i;

frmMain.edtTotal\_P.Text := IntToStr(i-1); // num P (Station)

frmMain.edtTotal\_S.Text := IntToStr(iNumS); // num S

end;

1. PLOT.DAT

//-------------------------

procedure PLOT\_dat(sPLOT\_dat : string);

var

i: Integer;

s, x: string;

listPlot\_dat: TStringList;

begin

listPlot\_dat := TStringList.Create;

try

with frmMain do begin

s := sgMainData.Cells[0, 0];

listPlot\_dat.Add(s);

for i := 1 to sgMainData.RowCount - 1 do begin

// if Trim(sgMainData.Cells[0, i]) = '' then break;

if Trim(frmMain.sgMainData.Cells[3, i]) = '\*' then continue;

//----- station

x := sgMainData.Cells[0, i];

s := x + ' ';

//----- P minute + second

if Trim(sgMainData.Cells[3, i]) = '' then

s := s + sgMainData.Cells[1, i] + ' ';

//----- S minute + second

if (Trim(sgMainData.Cells[4, i]) <> '') and (Trim(sgMainData.Cells[6, i]) = '')

then s := s + sgMainData.Cells[4, i]

else s := s + ' ';

//----- CODA

if Trim(sgMainData.Cells[7, i]) <> '' then

s := s + ' ' + sgMainData.Cells[7, i];

listPlot\_dat.Add(s);

end;

listPlot\_dat.Add('XXX');

listPlot\_dat.SaveToFile(lblProgramPath.Caption + 'Plotter\' + sPLOT\_dat);

end;

finally

listPlot\_dat.Free;

end;

end;

1. PHILSTA.DAT

procedure PHILSTA\_DAT(sPHILSTA\_dat: string; ProgramPath: string);

var

i: Integer;

s, StaName: string;

philstaList: TStringList;

FDConnection: TFDConnection;

FDQuery: TFDQuery;

DatabasePath, xLat, xLon, xElev: string;

begin

philstaList := TStringList.Create;

FDConnection := TFDConnection.Create(nil);

FDQuery := TFDQuery.Create(nil);

try

// Set up the database connection

DatabasePath := IncludeTrailingPathDelimiter(ProgramPath) + 'DB\PHILSTA.sdb';

FDConnection.DriverName := 'SQLite';

FDConnection.Params.Database := DatabasePath;

FDConnection.LoginPrompt := False;

FDConnection.Connected := True;

FDQuery.Connection := FDConnection;

for i := 1 to frmMain.sgMainData.RowCount - 1 do

begin

StaName := Trim(frmMain.sgMainData.Cells[0, i]);

if StaName = '' then Break;

// Query the database for latitude and longitude

FDQuery.SQL.Text := 'SELECT station\_code, latitude, longitude, elevation FROM Stations WHERE station\_code = :StaName LIMIT 1';

FDQuery.ParamByName('StaName').AsString := Copy(StaName, 1, 3);

FDQuery.Open;

if not FDQuery.Eof then

begin

xLat := FormatFloat('00.000', FDQuery.FieldByName('latitude').AsFloat);

xLon := FormatFloat('000.000', FDQuery.FieldByName('longitude').AsFloat);

xElev := FormatFloat('000', FDQuery.FieldByName('elevation').AsInteger);

s := StaName + ' ' + xLat + ' ' + xLon + ' ' +xElev;

// Add the station data to the list

philstaList.Add(s);

end;

FDQuery.Close;

end;

// Save the list to the specified file

philstaList.SaveToFile(IncludeTrailingPathDelimiter(ProgramPath) + 'Plotter\' + sPHILSTA\_dat);

finally

// Cleanup

philstaList.Free;

FDQuery.Free;

FDConnection.Free;

end;

end;