```
unit Settings;

{$mode objfpc}{$H+}

//========================================================================================
//
// Unit : Settings.pas
//
// Description : This module provides
//
// Called By : AppInit : ShowInit
//             MaintenanceTypes : TfrmMaintenanceTypes.DefaultRVFileExists
//             RVTypes : TfrmRVTypes.DefaultRVFileExists
//
// Calls : HUtils : ValidDirectoryCharacter
//
// Ver. : 1.00
//
// Date : 16 Mar 2015
//
//========================================================================================

interface

uses
  Classes, SysUtils, FileUtil, Forms, Controls, Graphics, Dialogs, Buttons,
  StdCtrls, INIFiles,
  // Application units
  HUtils;

type

  { TfrmSettings }

  TfrmSettings = class(TForm)
    bbtOK: TBitBtn;
    bbtCancel: TBitBtn;
    edtAppPath: TEdit;
    edtLogsPath: TEdit;
    edtDataPath: TEdit;
    stxAppPath: TStaticText;
    stxLogPath: TStaticText;
    stxLogsPath: TStaticText;
    procedure bbtCancelClick(Sender: TObject);
    procedure bbtOKClick(Sender: TObject);
    procedure edtDataPathKeyPress(Sender: TObject; var Key: char);
    procedure edtLogsPathKeyPress(Sender: TObject; var Key: char);
    procedure FormCreate(Sender: TObject);
    procedure FormShow(Sender: TObject);
  private
    { private declarations }
    fNewSettingsFile : boolean;
    fAppPath : string;
    fLogsPath : string;
    fDataPath : string;
    procedure SetAppPath ( value : string );
    procedure SetLogsPath ( value : string );
    procedure SetDataPath ( value : string );
  public
    { public declarations }
    const
      cstrDefLogsDirName = 'Logs';
      cstrDefDataDirName = 'Data';
    var
      vstrSettingsFileName : string;
    procedure ReadSettingsFile;
    procedure WriteSettingsFile;
  published
    { published porperties }
    property blnNewSettingsFile : boolean
      read fNewSettingsFile write fNewSettingsFile;
    property strAppPath : string
      read fAppPath write SetAppPath;
    property strLogsPath : string
      read fLogsPath write SetLogsPath;
      property strDataPath : string
      read fDataPath write SetDataPath;
  end;

var
  frmSettings: TfrmSettings;

  vstrOriginalLogsPath : string;
  vstrOriginalDataPath : string;

implementation

{$R *.lfm}

const

  cstrSettingsFileName = 'Settings.cfg';

  cstrSectionApplication = 'APPLICATION';
  cstrKeyNewSettingsFile = 'New Settings File';
  cstrKeyLogsPath = 'Logs Path';
  cstrKeyDataPath = 'Data Path';

var

  SettingsFile : TINIFile;

//==========================================================================================
//          PRIVATE ROUTINES
//==========================================================================================
procedure TfrmSettings.SetAppPath ( Value : string );
begin
  if Value <> fAppPath then
  begin
    fAppPath := Value + '\';
    edtAppPath.Text := fAppPath;
  end;// if Value <> fAppPath
end;// procedure TfrmSettings.SetAppPath

//==========================================================================================
procedure TfrmSettings.SetLogsPath ( Value : string );
begin
  if Value <> fLogsPath then
  begin
    fLogsPath := Value + '\';
    edtLogsPath.Text := fLogsPath;
  end;// if Value <> fLogsPath
end;// procedure TfrmSettings.SetLogsPath

//======================================================================================
procedure TfrmSettings.SetDataPath ( Value : string );
begin
  if Value <> fDataPath then
  begin
    fDataPath := Value + '\';
    edtDataPath.Text := fDataPath;
  end;// if Value <> fDataPath
end;// procedure TfrmSettings.SetDataPath

//======================================================================================
//          PUBLIC ROUTINES
//======================================================================================

//======================================================================================
//          SUPPPORT ROUTINES
//======================================================================================

//======================================================================================
//          MENU ROUTINES
//======================================================================================

//======================================================================================
//          FILE ROUTINES
//======================================================================================
procedure TfrmSettings.ReadSettingsFile;

var
  TStr : string;

begin

  SettingsFile := TINIFile.Create(vstrSettingsFileName);

  fNewSettingsFile := SettingsFile.ReadBool(cstrSectionApplication, cstrKeyNewSettingsFile,
                                                                 True);
  fLogsPath := SettingsFile.ReadString(cstrSectionApplication, cstrKeyLogsPath,
                                                                 fLogsPath);
  fDataPath := SettingsFile.ReadString(cstrSectionApplication, cstrKeyDataPath,
                                                                 fDataPath);

end;// procedure TfrmSettings.ReadSettingsFile

//======================================================================================
procedure TfrmSettings.WriteSettingsFile;
begin

  SettingsFile.WriteBool(cstrSectionApplication, cstrKeyNewSettingsFile, fNewSettingsFile);
  SettingsFile.WriteString(cstrSectionApplication, cstrKeyLogsPath, fLogsPath);
  SettingsFile.WriteString(cstrSectionApplication, cstrKeyDataPath, fDataPath);

  SettingsFile.UpdateFile;
  SettingsFile.Free;

end;// procedure TfrmSettings.WriteSettingsFile

```

```
//===========================================================================
//          COMMAND BUTTONS ROUTINES
//===========================================================================
procedure TfrmSettings.bbtOKClick(Sender: TObject);

const
  cbytMinLogsPathLength = 3;
  cbytMinDataPathLength = 3;

begin

  // Validate the settings

  // LogsPath
  if edtLogspath.Text <> vstrOriginalLogsPath then
  begin

    if Length(edtLogspath.Text) < cbytMinLogsPathLength then
    begin
      showmessage('Invalid LogsPath');
      ModalResult := MrNone;
      edtLogsPath.SetFocus;
      Exit;
    end;// if Length(edtLogspath.Text) < cbytMinLogsPathLength

    if (edtLogspath.Text[Length(edtLogspath.Text)] <> '\') then
      edtLogspath.Text := edtLogspath.Text + '\' ;

    fLogsPath:= edtLogspath.Text;

    // DataPath
    if Length(edtDatapath.Text) < cbytMinDataPathLength then
    begin
      showmessage('Invalid DataPath');
      ModalResult := MrNone;
      edtDataPath.SetFocus;
      Exit;
    end;// if Length(edtDataPath.Text) < cbytMinDataPathLength

    if (edtDataPath.Text[Length(edtDataPath.Text)] <> '\') then
      edtDataPath.Text := edtDataPath.Text + '\' ;

    fDataPath:= edtDataPath.Text;

  end;// if edtDataPath.Text <> fDataPath

end;// procedure TfrmSettings.bbtOKClick

//===========================================================================
procedure TfrmSettings.bbtCancelClick(Sender: TObject);
begin
  fLogsPath := vstrOriginalLogsPath;
  edtLogsPath.Text := fLogsPath;
end;// procedure TfrmSettings.bbtCancelClick

//===========================================================================
//          CONTROL ROUTINES
//===========================================================================

//===========================================================================
//          ON CLICK ROUTINES
//=========================================================================================

//=========================================================================================
//          ON CHANGE ROUTINES
//=========================================================================================

//=========================================================================================
//          KEYPRESS ROUTINES
//=========================================================================================
procedure TfrmSettings.edtLogsPathKeyPress(Sender: TObject; var Key: char);
begin
  Key := ValidDirectoryCharacter(Key);
end;// procedure TfrmSettings.edtLogsPathKeyPress

//=========================================================================================
procedure TfrmSettings.edtDataPathKeyPress(Sender: TObject; var Key: char);
begin
  Key := ValidDirectoryCharacter(Key);
end;// procedure TfrmSettings.edtDataPathKeyPress

//=========================================================================================
//          FORM ROUTINES
//=========================================================================================
procedure TfrmSettings.FormCreate(Sender: TObject);
begin

    vstrSettingsFileName := fAppPath + cstrSettingsFileName;

end;// procedure TfrmSettings.FormCreate

//=========================================================================================
procedure TfrmSettings.FormShow(Sender: TObject);
begin

  // Save the Original Data Elements
  vstrOriginalLogsPath := fLogsPath;
  edtLogsPath.Text := fLogsPath;

  vstrOriginalDataPath := fDataPath;
  edtDataPath.Text := fDataPath;

end;// procedure TfrmSettings.FormShow

//=========================================================================================

end.// unit Settings

```