

```

unit VLSSystemEntryPoints;

interface

uses SysUtils, LCRGlobals, VCL.FlexCel.Core, FlexCel.XlsAdapter;

type
  TVLSSystemEntryPoints = class
  public
    Connections: integer;
    NumberLSLs: integer;
    P90_base: double;
    Num_EP: integer;

    Xls: TExcelFile;

    constructor Create;
    destructor Destroy; override;

    procedure GetVLSEPData(pwsid: string);
  private
    //Xls: TExcelFile;
  end;

implementation

uses System.Variants;

{ TVLSSystemEntryPoints }

constructor TVLSSystemEntryPoints.Create;
var
  filename: string;
begin
  filename := DataPath + 'VLEntryPointValues.xlsx';

  Xls := TXlsFile.Create(Filename, False);
  Xls.ActiveSheetByName := 'Inputs';
  {
    VLEntryPointValues.xlsx
    1 PWSID
    2 p_b3
    3 p_b3_r
    4 Baselineph_wocct
    5 Baselineph_wph
    6 Baselineph_wpo4ph
    7 Baselineph_woph
    8 BaselineP04Dose
    9 CCT
    10 LSL_vls
  }

```

```

    11 num_lsl_known_vls
    12 Num_lsl_unknown_lead_vls
    13 Num_lsl_unknown_vls
    14 Num_lsl_pws_vls
    15 NumberEPs
    16 pbasephpo4
    17 pbaseph
    18 pbasepo4
    19 Num_connect_vls
    20 Population
    21 Recno
}

end;

destructor TVLSSystemEntryPoints.Destroy;
begin
    FreeAndNil(Xls);

    inherited;
end;

procedure TVLSSystemEntryPoints.GetVLSEPDData(pwsid: string);
var
    r: integer;
    bFound: boolean;
begin
    bFound := false;

    for r := 2 to Xls.RowCount do
        begin
            if Xls.GetStringFromCell(r,1) = Copy(pwsid,1,9) then
                begin
                    Connections := xls.GetCellValue(r, 2).AsVariant;

                    if VarIsNumeric(xls.GetCellValue(r, 3).AsVariant) then
                        NumberLSLs := xls.GetCellValue(r, 3).AsVariant
                    else
                        NumberLSLs := -1;

                    P90_base := xls.GetCellValue(r, 4).AsVariant;
                    Num_EP := xls.GetCellValue(r, 5).AsVariant;
                    bFound := true;
                    break;
                end;
            end;
        end;

    if not bFound then
        csl('Row not found in VLSEntryPointsValues.xlsx for ' + pwsid);
    //    raise Exception.Create('Row not found in VLSEntryPointsValues.xlsx for ' +

```

```
pwsid);  
end;
```

```
end.
```