

StateStrataData.pas

```
unit StateStrataData;

interface

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

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

    constructor Create;
    destructor Destroy; override;

    procedure GetSystemData(pwsid: string);
  private
    Xls: TExcelFile;
  end;

implementation

{ TStateStrataData }

constructor TStateStrataData.Create;
var
  filename: string;
begin
  filename := DataPath + 'StateStrataData.xlsx';

  Xls := TXlsFile.Create(Filename, False);
  Xls.ActiveSheetByName := 'Inputs';
  {
    StateStrataData.xlsx
    1 PWSID
    2 Connections
    3 NumberLSLs
    4 P90_base
    5 Num_EP
  }
end;

destructor TStateStrataData.Destroy;
begin
  FreeAndNil(Xls);
```

StateStrataData.pas

```
    inherited;
end;

procedure TStateStrataData.GetSystemData(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;
                    NumberLSLs := xls.GetCellValue(r, 3).AsVariant;
                    P90_base := xls.GetCellValue(r, 4).AsVariant;
                    Num_EP := xls.GetCellValue(r, 5).AsVariant;
                    bFound := true;
                    break;
                end;
            end;
        end;

    if not bFound then
        raise Exception.Create('Row not found in VLSSystemData.xlsx for ' + pwsid);
    end;
end.
```