**Pascal Script Manual for MicroGisEditor**

**Types**

**TPoint**  declared as

TPoint = record

X:Integer;

Y:Integer;

End;

**TDoublePoint** declared as

TDoublePoint = record

X:double;

Y:double;

End;

**Functions:**

**function Random(x:integer):integer;**

Generates random numbers within a specified range.

**function IntPower(const Base: Single; const Exponent: Integer): Single;**

Calculates the integral power of a base value.

**function IntToStr(const iValue: integer):string;**

Converts an integer to a string.

**function IntToHex(Value: Integer; Digits: Integer): string;**

Returns the hex representation of an integer.

**function StrLength(Str: string): integer;**

Returns the length of String.

**function SubString(Str: string; iBegin: Integer; iCount: Integer): string;**

Returns the substring of a string.

**function strRound(Value: Double; Precision: Integer): string;**

Returns the value of Value rounded to Float, were Precision – numbers of digitals after point and converted to String.

**Procedures:**

**procedure SetProjection(const \_iAxisA:integer; const \_iAxisB:integer);**

This procedure NOT USE at DAT file!!! Use only for debug your script! Procedure set axis of ellipsoid (sphere) for debug Only!

**procedure Tile2LonLatSphere(const pntTilePos: TPoint; Zoom: byte** **; var pntOut :TDoublePoint);**

Сalculates Longitude And Latitude as TDoublePoint from Tile coordinates , for Mercator on Sphere

**procedure Tile2LonLatSimple(const pntTilePos: TPoint;Azoom: byte** **; var pntOut :TDoublePoint);**

Сalculates Longitude And Latitude as TDoublePoint from Tile coordinates , for geodetic coordinates

**procedure Tile2LonLatEllipsoid(const AXY: TPoint; Azoom: byte** **; var pntOut :TDoublePoint);**

Сalculates Longitude And Latitude as TDoublePoint from Tile coordinates , for for Mercator on Ellipsoid

**procedure LonLat2MetersSphere(const pntLonLat: TDoublePoint** **; var pntOut :TDoublePoint);**

Сalculates tile position on meters as TDoublePoint from Tile coordinates , for Mercator on Sphere

**procedure LonLat2MetersSimple(const pntLonLat: TDoublePoint** **; var pntOut :TDoublePoint);**

Сalculates tile position on meters as TDoublePoint from Tile coordinates , for geodetic coordinates

**procedure LonLat2MetersEllipsoid(const ALl: TDoublePoint** **; var pntOut :TDoublePoint);**

Сalculates tile position on meters as TDoublePoint from Tile coordinates , for Mercator on Ellipsoid

**DEBUG EXAMPLE**

Script

var

pntTemp : TPoint;

TopLeftM,BottomRightM,pntdblTemp : TDoublePoint;

begin

SetProjection(6378137,6356752); //USE ONLY FOR DEBUG!!!

//USE ONLY FOR DEBUG!!!

DefaultURL:='http://78.46.61.141/cgi-bin/tilecache-2.11/tilecache.py?LAYERS=topomapper\_gmerc&FORMAT=image%2Fjpeg&SERVICE=WMS&VERSION=1.1.1&REQUEST=GetMap&STYLES=&EXCEPTIONS=application%2Fvnd.ogc.se\_inimage&SRS=EPSG%3A900913&WIDTH=256&HEIGHT=256&BBOX='

//USE ONLY FOR DEBUG!!!

X:=0;Y:=0;Z:=0;

//Script begin

pntTemp.x:=X;

pntTemp.y:=Y;

Tile2LonLatEllipsoid(pntTemp,Z, pntdblTemp);

LonLat2MetersEllipsoid(pntdblTemp, TopLeftM);

pntTemp.x:=(X+1);

pntTemp.y:=(Y+1);

Tile2LonLatEllipsoid(pntTemp,Z, pntdblTemp);

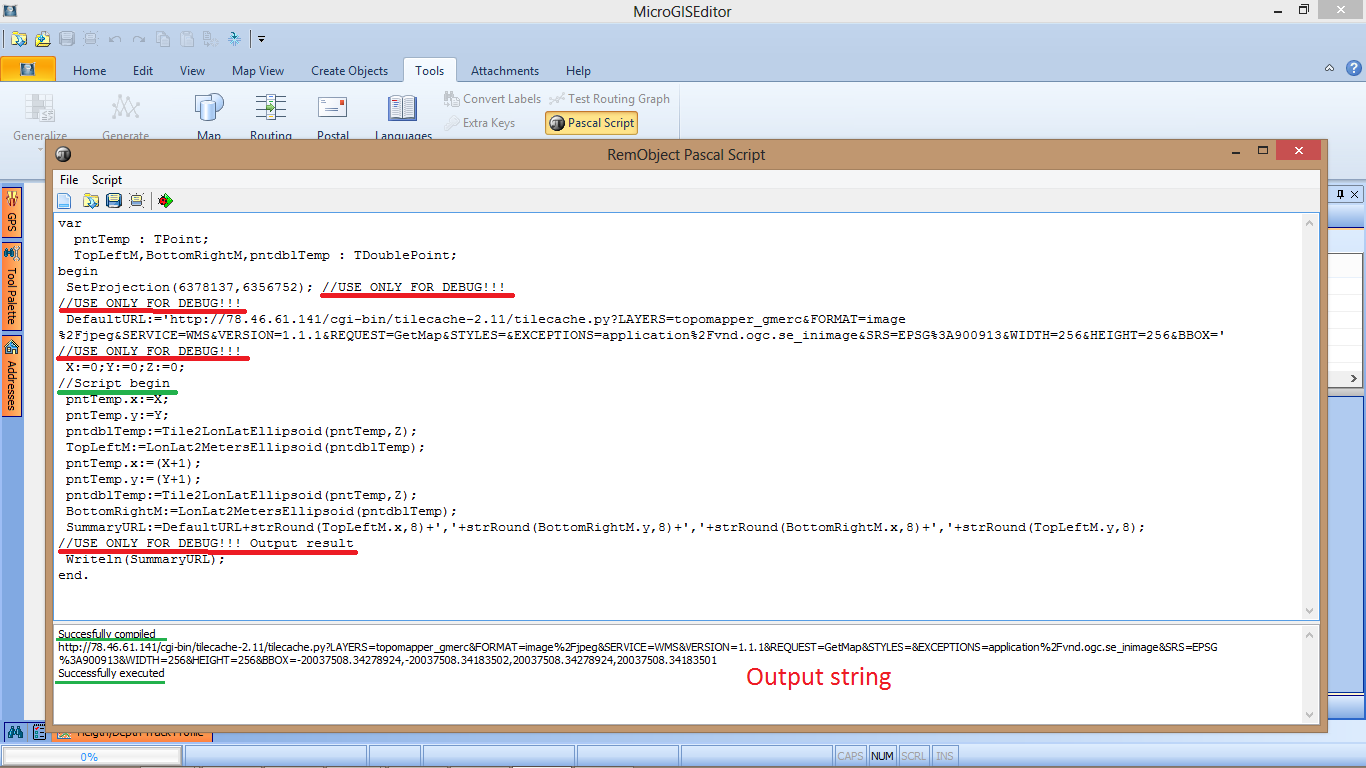
LonLat2MetersEllipsoid(pntdblTemp, BottomRightM);

SummaryURL:=DefaultURL+strRound(TopLeftM.x,8)+','+strRound(BottomRightM.y,8)+','+strRound(BottomRightM.x,8)+','+strRound(TopLeftM.y,8);

//USE ONLY FOR DEBUG!!! Output result

Writeln(SummaryURL);

end.



**After debug Your result script:**

var

pntTemp : TPoint;

TopLeftM,BottomRightM,pntdblTemp : TDoublePoint;

begin

pntTemp.x:=X;

pntTemp.y:=Y;

Tile2LonLatEllipsoid(pntTemp,Z, pntdblTemp);

LonLat2MetersEllipsoid(pntdblTemp, TopLeftM);

pntTemp.x:=(X+1);

pntTemp.y:=(Y+1);

Tile2LonLatEllipsoid(pntTemp,Z, pntdblTemp);

LonLat2MetersEllipsoid(pntdblTemp, BottomRightM);

SummaryURL:=DefaultURL+strRound(TopLeftM.x,8)+','+strRound(BottomRightM.y,8)+','+strRound(BottomRightM.x,8)+','+strRound(TopLeftM.y,8);

end.

**Section of DAT file**

[WEB MAP] //Must be first

Name= //Name of service

Cache= //SubFolder for cache

DefaultURL=//Part of URL address, which is const

Ext= // Extention of tiles - PNG or JPG

Projection= //Projection 0-Mercator on Ellipsoid, 1-Mercator on sphere, 3 –geodetic coordinates

Glyph=//Picture BMP for menu

AxisA=//Axis A for datum

AxisB=//Axis b for Datum

MaxLevel=//Max level for service (can be deleted)

DefHotKey=//Hot key

[SCRIPT] //Script section – must be last!

var

pntTemp : TPoint;

TopLeftM,BottomRightM,pntdblTemp : TDoublePoint;

begin

pntTemp.x:=X;

pntTemp.y:=Y;

Tile2LonLatEllipsoid(pntTemp,Z, pntdblTemp);

LonLat2MetersEllipsoid(pntdblTemp, TopLeftM);

pntTemp.x:=(X+1);

pntTemp.y:=(Y+1);

Tile2LonLatEllipsoid(pntTemp,Z, pntdblTemp);

LonLat2MetersEllipsoid(pntdblTemp, BottomRightM);

SummaryURL:=DefaultURL+strRound(TopLeftM.x,8)+','+strRound(BottomRightM.y,8)+','+strRound(BottomRightM.x,8)+','+strRound(TopLeftM.y,8);

end.