Migration of SMD spatial information into Sola database

Author: Elton Manoku

Version: 1.0

Date: May 31, 2012

# Introduction

This document describes the process of migrating the spatial information produced in SMD to Sola Database.

# Spatial information of SMD

## Reference coordinative system

All current spatial information in SMD is in ‘War Office’ coordinative system.

The EPSG code: **2136**.

## Parcel identifier

A parcel in SMD is identified countrywide by the combination of:

* Region

*Example: Greater Accra identified by a code GA*

* Registration district (These are districts differing from the administrative districts used for other purposes)

*Example: 03*

* Section (Sections are unique in the whole region)

*Example: 021*

* Block (The block is unique within a section)

*Example: 4*

* Parcel / Lot number (The parcel number is unique within a block)

*Example: 45*

So a parcel identifier would look like: GA/03/021/4/45. Because the section number is unique within a region also this identifier is valid: GA/021/4/45. But it is common to refer to a section within a certain district.

## Base objects

With the base objects are meant the spatial features that are used to divide the country territory without the parcel object. So more concrete:

* Region
* Registration district
* Section
* Block

This information will not change often or not at all so it can be seen as static information.

From all these objects, Section and Block can be retrieved from the digital archive.

## Digital archive

From this information SMD maintains its spatial information in the level of blocks and parcels.

The spatial information that is produced/ archived in SMD consists of Zip files named after the section number that they belong to.

*Example:*

*All spatial information about the section 20 is stored in a ZIP archive called S020.zip.*

Each archive file contains two files:

BLOCK.E00: This holds the spatial information about blocks.

LOT.E00: This holds the spatial information about parcels or lots.

The section nr is unique throughout the region. So, for the Greater Accra, all section files can be placed in the same folder.

# Spatial information in Sola Database

Sola Database stores the spatial data in tables where one of the columns is of the type geometry.

## Reference coordinative system

For the future coordinate system is desired to be UTM Zone 30N. The EPSG code: **32630**.

## SMD hierarchical objects in Sola Database

Sola Database schema offers the possibility to store spatial data that describe the hierarchical structure of regions, registration districts, sections, blocks in table: **cadastre.spatial\_unit\_group**.

This table can accommodate all these hierarchical data by using the **hierarchy\_level**.

The levels of hierarchy can be defined as below.

|  |  |
| --- | --- |
| **Type of data** | **Hierarchy** |
| Region | 1 |
| Registration district | 2 |
| Section | 3 |
| Block | 4 |

## SMD parcels in Sola Database

Parcels in Sola Database are considered cadastre objects of type ‘parcel’. This is the table **cadastre.cadastre\_object**.

# Process of migration

The process of migration explains the steps and tools used for each step.

## Assumptions

### Source

It is assumed that the archive files are found in a source folder and they are named according to the template S<section nr>.zip.

*Example: Archive of the section 20 is: S020.ZIP*.

### Destination

The destination database must be created according to Sola guidelines. It must contain a schema called **staging\_area**.

Inside this schema should be a table called **shape\_block** and a table called **shape\_lot**. These tables can be created with the script below:

CREATE TABLE staging\_area.shape\_block

(

gid serial NOT NULL,

area double precision,

perimeter double precision,

"cov\_block#" integer,

"cov\_block-" integer,

blockno smallint,

x\_coord double precision,

y\_coord double precision,

geom geometry,

section character varying(20),

region character varying(5),

CONSTRAINT shape\_block\_pkey PRIMARY KEY (gid ),

CONSTRAINT enforce\_dims\_geom CHECK (st\_ndims(geom) = 2),

CONSTRAINT enforce\_geotype\_geom CHECK (geometrytype(geom) = 'MULTIPOLYGON'::text OR geom IS NULL),

CONSTRAINT enforce\_srid\_geom CHECK (st\_srid(geom) = 2136)

);

CREATE TABLE staging\_area.shape\_lot

(

gid serial NOT NULL,

area double precision,

perimeter double precision,

"cov\_lot#" integer,

"cov\_lot-id" integer,

lotno smallint,

"desc" character varying(65),

noaccess smallint,

numpts smallint,

x\_coord double precision,

y\_coord double precision,

geom geometry,

CONSTRAINT shape\_lot\_pkey PRIMARY KEY (gid ),

CONSTRAINT enforce\_dims\_geom CHECK (st\_ndims(geom) = 2),

CONSTRAINT enforce\_geotype\_geom CHECK (geometrytype(geom) = 'MULTIPOLYGON'::text OR geom IS NULL),

CONSTRAINT enforce\_srid\_geom CHECK (st\_srid(geom) = 2136)

);

## Converting ArcINFO files into staging area

This conversion will transform the spatial data about section, block and parcel (lot) and insert them into a staging area in the Sola Database.

### Step 1 – Unpack the archive

**Windows**: Z7.exe

**Ubuntu**: Install package unzip: sudo apt-get install unzip

After unzipping, files BLOCK.E00 and LOT.E00 are found in destination.

### Step 2 – Convert block.e00 and lot.e00 to ArcInfo coverage format

**Windows**: avcimport.exe. This is freeware utility to convert ArcInfo Format to ArcInfo Coverage format

**Ubuntu**: Install package avce00: sudo apt-get install avce00

After converting, to the destination folder will be found many files of extension .adf.

### Step 3 – Convert ArcInfo Coverage to Shapefiles

**Windows**: ogr2ogr.exe: This is an open source utility which converts spatial data from one format to another.

**Ubuntu**: There are also versions for Linux. <http://www.gdal.org/ogr2ogr.html>

After conversion, in the destination folder you will find many shapefiles. The important shapefile, both for lot and block is: PAL.shp.

### Step 3 – Convert shapefiles to Postgresql insert statements scripts

**Windows**: shp2pgsql.exe. This is a postgis utility.

**Ubuntu**: The same as in Windows.

After conversion, .sql files will be generated with the necessary insert commands to load the shapefile into postgresql.

### Step 4 – Run insert statements scripts against the target database

**Windows**: psql.exe. This is a client utility of postgresql that is used to run scripts and commands against a database.

**Ubuntu**: The same as in Windows.

### Step 5 – Update the new records from the block with values of region and section

After the block records are loaded into table staging\_area.shape\_block, they have the values of columns region and section null. These values are filled with an update statement immediately after the new block records are added in the table. In section is filled the name of the source file (example: S020). In region is predefined in the beginning for which region this script will run (example: GA).

**Windows** and **Ubuntu**: The same as step 4

## Check, transform and load data from staging area into the Sola cadastre schema

This step consists in transforming the geometries to reference system of destination which is UTM and moving them into the tables of cadastre schema.