

# MapInfo, GDAL ja ogr2ogr

## ***Esimerkinä pääkaupunkiseudun seutukartta 2012***

*Viimeksi muutettu 13. helmikuuta 2012*

### ***Lyhennelmä***

GDAL-apuohjelmalla ogr2ogr voidaan muuntaa suomalaisia Mapinfo-muotoisia aineistoja muihin tiedostomuotoihin ja tietokantoihin, mutta muunnoksessa on otettava huomioon seuraavat asiat:

- GDAL:in Mapinfo-ajuri ei hallitse muunnoksia erilaisten merkistökoodausten välillä, joten muunnos esimerkiksi ISO-8859-1 -koodauksen ja UTF-8 -koodauksen välillä on tehtävä välivaiheen kautta
- Mapinfo-muoto, toisin kuin esimerkiksi shapefile-muoto, sallii pisteiden, viivojen ja alueiden tallentamisen samalla karttatasolle. Myös ominaisuustietojen nimet voivat Mapinfo-muodossa olla pidempiä kuin shapefile-muodossa. Mikäli haluaa säilyttää Mapinfo-tiedoston rakenteen muunnoksessa, niin on varmintä tehdä muunnos johonkin vastaavia ominaisuuksia tukevaan tiedostomuotoon, esimerkiksi GML:ään, tai tietokantaan, kuten Oracleen, PostGIS:iin tai Spatialiteen.
- Muunnokseen tarvittava laskenta-aika on noin 2 minuuttia, mutta valmistelevat työt vaativat oman aikansa.
- Tarvitaan GDAL-versio 1.9.0, jotta koordinaattijärjestelmästä toiseen muuntaminen onnistuu oikein pelkkiä EPSG-koodeja käyttämällä.

### ***Pääkaupunkiseudun seutukartta-aineisto ja sen muuntaminen ogr2ogr-ohjelmalla***

Pääkaupunkiseudun seutukartta-aineisto <http://www.hri.fi/fi/data/seutukartta/> tuli saataville tammikuussa 2012. Aineistoon kuuluu 60 karttatasoa, joista jokainen on omana Mapinfo-muotoisena neljän tiedoston tiedostonippuna, joten jakeluun sisältyy 240 tiedostoa.

Jokainen karttataso sisältää vain yhden tyyppisiä geometrioita (pisteitä, viivoja tai alueita). Osa viivatasoista sisältää kuitenkin sekä LineString- että MultiLineString-tyyppejä ja vastaavasti osa aluetasosta sekä Polygon- että MultiPolygon-tyyppejä. Aineiston käyttämä koordinaattijärjestelmä on KKJ:n 2-kaista (EPSG:2392).

Aineiston merkistökoodauksena on ”WindowsLatin1”. Koodin ”ISO-8859-1” käyttäminen tuotti käytännössä oikean tuloksen, vaikka teoriassa se ei ehkä tarkoita ihan samaa asiaa.

Ominaisuustietojen arvoissa esiintyy ei ASCII-merkkejä ja muutamilla karttatasoilla myös niiden nimissä. Ominaisuustietokenttien nimet voivat myös olla pitkiä, esimerkiksi ”metroasema” tai ”vesistö\_nimi\_r”.

Seuraavassa esimerkissä seutukartta-aineiston kaikki 60 karttatasoa muunnetaan yhdeksi Spatialite-tietokannaksi. Muunnoksen jälkeen koko aineisto on yhtenä tiedostona 240 tiedoston sijasta. Spatialite-muodossa on mahdollista säilyttää myös pitkät ominaisuustietojen nimet. Spatialite-muodon käyttämisen takia on kuitenkin otettava huomioon kaksi asiaa:

- Spatialite-tietokannat käyttävät UTF-8 -merkistökoodausta, joten muunnoksessa on tehtävä muunnos ISO-8859-1 koodauksesta UTF-8:aan.
- Spatialite-kanta itsessään tukee eri geometriatyyppeiden tallentamista samaan tauluun, mutta

monet yleisesti käytettävät Spatialite-tietokantaa tukevat sovellukset eivät tätä tue. Siksi on parasta tallentaa tason kaikki kohteet MultiLineString-muodossa, jos lähtöaineistossa on sekä LineString- että MultiLineString-tyypin kohteita. Vastaavasti sekalaisia aluekohteita sisältävien tasojen kohteet kannattaa muuntaa kaikki MultiPolygon-tyyppisiksi.

## **KKJ-2 -kaistassa olevan, ISO-8859-1 -merkistökoodausta käyttävän Mapinfo TAB muotoisten aineistojen muuntaminen ETRS-TM35FIN -järjestelmässä olevaksi Spatialite-tietokannaksi, joka käyttää UTF-8 -merkistökoodausta**

### ***Mallisuoritus 1:***

#### **Vaihe 1: Ei-ASCII-merkkien poistaminen skeemasta**

Tämä vaihe ei ole välttämätön, mutta koska joillakin ohjelmilla tunnetusti on ongelmia, jos ominaisuustietojen nimissä on ei-ASCII -merkkejä, niin kannattaa ainakin harkita niiden muuttamista vain ASCII-merkkejä sisältäviksi. Suomessa kun ollaan, niin tämä muunnos voi tuntua ärsyttävältä, joten harkinnan jälkeen ne voi päättää jättää ennalleenkin. Tässä tapauksessa lue myös mallisuoritus 2.

Ominaisuustietokenttien nimet voidaan vaihtaa avaamalla tekstinkäsittelyohjelmalla .TAB -päätteinen tiedosto, tässä esimerkiksi seutukartta-aineistoon kuuluva "m\_jarvet.tab"

```
!table
!version 300
!charset WindowsLatin1

Definition Table
  Type NATIVE Charset "WindowsLatin1"
  Fields 3
    Vesistö_nimi_s Char (40) Index 1 ;
    Vesistö_nimi_r Char (40) Index 2 ;
    Mtryhm Integer Index 3 ;
```

Muutetaan kntän nimet seuraavasti, ja tallennetaan muokattu TAB-tiedosto.

```
Vesisto_nimi_s Char (40) Index 1 ;
Vesisto_nimi_r Char (40) Index 2 ;
```

#### **Vaihe 2: Muunnetaan KKJ-2 -kaistassa oleva Mapinfo-tiedosto ETRS-TM35FIN -järjestelmää käyttäväksi GML-tiedostoksi**

Käytetään ogr2ogr-ohjelmaa ja annetaan seuraava komento:

```
ogr2ogr -f GML -t_srs epsg:3067 m_jarvet.GML m_jarvet.TAB
--config OGR_FORCE_ASCII no
```

**Huom!** Jos käytössä on GDAL-versio 1.9.0 tai uudempi voidaan muunnos KKJ2->ETRS-TM35FIN eli EPSG:2392 -> EPSG:3067 tehdä kuten yllä on esitetty. Vanhemmilla GDAL-versioilla tämä johtaa projektiovirheeseen. Tätä aihetta käsitellään Latuviitan ohjeessa [Tee muunnokset KKJ<->TM35FIN oikein](#)

### Vaihe 3: Korjataan GML-tiedoston merkistökoodi oikeaksi

GDAL:in Mapinfo-ajuri ei hallitse erilaisia merkistökoodauksia, vaan se olettaa, että koodaus on aina UTF-8. Kuten edellä olevasta .TAB -tiedoston alusta nähdään, seutukartoissa käytetään merkistöä WindowsLatin1, jota vastaa koodi ISO-8859-1. Siitä syystä avataan edellä luotu GML-tiedosto ja korjataan sen ensimmäisellä rivillä oleva merkistökoodi oikeaksi.

Ennen korjausta

```
<?xml version="1.0" encoding="utf-8" ?>
```

Korjauksen jälkeen

```
<?xml version="1.0" encoding="iso-8859-1" ?>
```

Tallennetaan korjattu GML-tiedosto. Koska GDAL:in GML-ajuri tukee muutamia erilaisia merkistökoodauksia, niin se muuntaa seuraavassa vaiheessa myös kaikki merkit ISO-8859-1-koodauksesta UTF-8:aan. Lisätietoa tästä GML-ajurin ohjesivulla

[http://gdal.org/ogr/drv\\_gml.html](http://gdal.org/ogr/drv_gml.html)

### Vaihe 4: Muunnetaan GML-tiedosto Spatialite-muotoon

```
ogr2ogr -f SQLite -dsco spatialite=yes -dsco init_with_epsg=yes -gt 2048 -a_srs epsg:3067 pk_seutukartta.sqlite A_es_pie.GML -nlt polygon -lco LAUNDER=yes
```

Parametrien merkitys

<b>-f SQLite</b>	Kohteena on SQLite-tietokanta
<b>-dsco spatialite=yes</b>	Kohteena on nimenomaan Spatialite-tyylinen SQLite-tietokanta
<b>-dsco init_with_epsg=yes</b>	Kirjoitetaan syntyvään tietokantaan kaikkien tuettujen koordinaattijärjestelmien määrittelyt
<b>-gt 2048</b>	Kirjoitetaan 2048 kohdetta ennen Commit-toimintoa; tämä nopeuttaa muunnosta verrattuna oletusarvon 200 käyttämiseen
<b>-a_srs epsg:3067</b>	Liitetään aineistoon EPSG-koodi 3067
<b>-nlt multipolygon</b>	Koska tiedetään, että osa järvistä on multipolygoneja, niin kirjoitetaan kaikki järvet tietokantaan multipolygoneina.

Lisää tietoa Spatialite-ajurista ja sen parametreista sivulla

[http://gdal.org/ogr/drv\\_sqlite.html](http://gdal.org/ogr/drv_sqlite.html)

### Vaihe 5: sama uudelleen

Toistetaan vaiheet 1, 2, 3 ja 4 lopuille 59 karttatasolle. Erotuksena on ainoastaan se, että Spatialite-kanta on jo olemassa, joten 4-vaiheessa ei luoda uutta tietokantaa vaan päivitetään jo olemassa olevaa. Komento on tässä tapauksessa Tarvittava komento on silloin tällainen:

```
ogr2ogr -f SQLite -update -gt 2048 -a_srs epsg:3067 pk_seutukartta.sqlite A_es_suu.GML -nlt polygon -lco LAUNDER=yes
```

## Huomautuksia mallisuoritukseen 1.

Edellä vaiheessa 3 määrättiin luotavalle uudelle tietokantataululle tarkka geometriatyyppi -ln -parametrin avulla. Mutta kuinka oikea geometriatyyppi selvitetään? Yksi vaihtoehto on jättää se selvittämättä ja antaa kaikille viivatasoille tyypiksi ”MULTILINESTRING” ja kaikille aluekohteille ”MULTIPOLYGON”. Mutta seutukartta-aineistosta ei aina voi esimerkiksi nimen perusteella päätellä, mitä geometriota tasolle kuuluu. Esimerkiksi tasolla ”l\_kiitor” on kiitoradat alueina, vaikka arvaamalla voisi päätellä niiden olevan viivoja. Ogrinfo-työkalustakaan ei ole apua, koska se listaa vain, että taso sisältää ylipäätään geometrioita:

```
ogrinfo -so -al l_kiitor.tab
Had to open data source read-only.
INFO: Open of `l_kiitor.tab'
      using driver `MapInfo File' successful.
```

```
Layer name: l_kiitor
Geometry: Unknown (any)
```

Mikä siis avuksi? Yksi tapa on tehdä ensin alustava muunnos ogr2ogr-ohjelmalla Spatialite-muotoon yrittämättäkään määrätä syntyvän tietokantatason geometriatyyppejä. Tietokanta voidaan sitten avata Spatialite-gui -ohjelmalla, jossa on oma erityinen työkalu geometriatyypien tarkastamiseen.

## **Mallisuoritus 2:**

Edellisessä esimerkissä poistettiin ensin erikoismerkit Mapinfo-tiedoston skeemasta muokkaamalla .tab -tiedostoa, tehtiin väliaikainen muunnos GML-muotoon ja korjattiin käsin muokkaamalla GML-tiedoston merkistökoodi oikeaksi. Seuraavassa esimerkissä esitetään, mitä tehdä jos skeemaan kuitenkin halutaan ominaisuustietojen nimiin mukaan ei-ASCII-merkkejä.

### **Vaihe 1: Muunnos Mapinfo-muodosta GML-muotoon.**

Ensimmäinen vaihe voidaan tehdä suoraan ogr2ogr-ohjelmalla mutta TAB-tiedostoja ei muokata käsin ennen muunnoksen tekemistä vaan ominaisuustietojen nimiin mahdollisesti kuuluvien erikoismerkkien annetaan olla paikallaan. Komento on sama kuin edellisessäkin esimerkissä

```
ogr2ogr -f GML -t_srs epsg:3067 m_jarvet.GML m_jarvet.TAB
--config OGR_FORCE_ASCII no
```

### **Vaihe 2: GML-tiedoston merkistökoodauksen korjaus ja .xsd-tiedoston tuhoaminen**

Samoin kuin edellisessä esimerkissä korjataan GML-tiedoston ensimmäiselle rivillä oikea merkistökoodaus

Ennen korjausta

```
<?xml version="1.0" encoding="utf-8" ?>
```

Korjauksen jälkeen

```
<?xml version="1.0" encoding="iso-8859-1" ?>
```

Ogr2ogr luo oletusasetuksia käytettäessä GML-muunnoksen yhteydessä myös skeemamäärittelytiedoston, jonka nimen alkuosa on sama kuin GML-tiedostolla mutta tarkentimena on .xsd. Tämä tiedosto tuhoetaan ennen kuin siirrytään seuraavaan vaiheeseen.

### Vaihe 3: GML-tiedoston muuntaminen Spatialite-muotoon

Tämä vaihe on sama kuin edellisessäkin esimerkissä, ja sekä ominaisuustietojen nimiin että niiden sisällössä mahdollisesti olevat ei-ASCII-merkit muuntuvat oikein

```
ogr2ogr -f SQLite -dsco spatialite=yes -dsco init_with_epsg=yes -gt 2048 -a_srs epsg:3067 pk_seutukartta.sqlite A_es_pie.GML -nlt polygon -lco LAUNDER=yes
```

#### Huomautuksia mallisuoritukseen 2.

GDAL:in GML-ajuri kirjoittaa oletusasetuksilla sekä GML-tiedoston että sitä vastaavan skeematiedoston (.xsd). GML-ajuri tukee muutamaa merkistökoodausta (US-ASCII, UTF-8, UTF-16 ja ISO-8859-1). Yllättävää kyllä, ajuri tukee niitä kuitenkin vain GML-tiedoston käsittelyssä, mutta ei skeematiedoston käsittelyssä. Tämä johtuu siitä, että skeematiedoston XML-sisällön jäsentämiseen käytetään eri ohjelmakirjastoa.

Kun edellisen mallisuoritus 2:n vaiheessa 2 tehtiin muunnos Mapinfosta GML-muotoon, niin GDAL kirjoitti varsinaisen tietosisällön .gml-tiedostoon ja tiedon rakennekuvauksen, joka sisältää mm. ominaisuustietokenttien nimet .xsd-tiedostoon. GDAL kirjoittaa molempien tiedostojen otsikkoriville, että niissä käytetään merkistökoodausta UTF-8, mikä on esimerkin tapauksessa väärin. GML-tiedoston merkistökoodaus voidaan korjata oikeaksi muokkaamalla otsikkoriviä niin että siinä on todellisen koodauksen tunnus ISO-8859-1, mutta vaikka sama muutos tehtäisiin .xsd-tiedostoon, niin sillä ei kuitenkaan olisi mitään vaikutusta. GDAL olettaa joka tapauksessa, että se on UTF-8-koodattu. Tästä seuraa, se, jos lähtöaineiston ominaisuustietojen nimissä on ei-ASCII-merkkejä, niin ne muuntuvat väärin UTF-8-muotoon. Ensimmäisessä malliesimerkissä tätä ei tapahtunut, koska alkuperäisiä Mapinfo-tiedostoja muokattiin niin, että missään ominaisuustietojen nimissä ei ollut muita kuin ASCII-merkkejä.

On ehkä yllättävää, että helpoin tapa ominaisuustietojen nimien muunnos UTF-8-muotoon tapahtumaan oikein on tuhota .xsd-tiedosto. GDAL:in GML-ajuri nimittäin toimii niin, että jos skeematiedostoa ei ole olemassa, niin varsinainen GML-datatiedosto käydään ensin läpi alusta loppuun ja GDAL luo skeematiedoston sen perusteella, mitä elementtejä tiedoista löytyy. Koska gml-tiedoston käsittelyssä ISO-8859-1 koodaus hallitaan, niin GDAL selviytyy myös ei-ASCII-merkkejä sisältävistä ominaisuustietojen nimistä.

#### **GDAL:in merkistökoodauksesta**

Yhteenveto siitä, mitä vaikeuksia on ISO-8859-1-merkistöä käyttävien Mapinfo-tiedostojen muuntamiseksi muihin tiedostomuotoihin GDAL:in avulla.

- GDAL:in Mapinfo-ajuri ei hallitse eri merkistökoodauksia vaan merkit otetaan sellaisina kuin ne ovat ja siirretään ilman muunnoksia tulostiedostoihin.
- GDAL:ssa ei muutenkaan ole yleistä, kaikille ajureille yhteistä menetelmää merkistökoodauksen hallintaan
- GDAL:in GML-ajuri hallitsee seuraavien merkistöjen väliset muunnokset: US-ASCII, UTF-8, UTF-16 ja ISO-8859-1. Merkistötuki koskee kuitenkin vain varsinaista gml-tiedostoa, ei .xsd-skeematiedostolle
- GDAL tunnistaa ei-ASCII-merkit ja oletusasetuksilla pakottaa ne ASCII-merkeiksi. Tämä tuottaa sinänsä käyttökelpoisen tuloksen, mutta jos halutaan saada merkit muunnetuksi oikein esimerkiksi UTF-8-koodatuiksi, niin tämä pakkomuunnos on kiellettävä antamalla org2ogr-

ohjelmalla parametri ”--config OGR\_FORCE\_ASCII NO”

- GML-muunnoksessa tulokseksi syntyy .gml- ja .xsd-tiedostot, jotka on otsikossa merkitty käyttämään UTF-8-koodausta, vaikka sisältö oikeasti onkin lähtöaineiston mukainen eli tässä tapauksessa ISO-8859-1
- Gml-tiedosto on helppo korjata kuntoon muokkaamalla sen otsikkoriviä ja kirjoittamalla siihen oikea merkistökoodi, mutta koska GDAL ei tue eri merkistöjä .xsd-tiedoston tulkittamisessa niin tämä korjaus ei riitä
- Pienellä tempulla muunnos saadaan onnistumaan, eli GDAL voidaan ohjata luomaan skeema tutkimalla gml-tiedosto läpikotaisin sen merkistökoodauksen ymmärtävälle menetelmällä. GDAL käyttää tätä polkua, jos se ei löytä .xsd-tiedostoa, joten .xsd-tiedosto voidaan yksinkertaisesti tuhota tai sitten sen syntyminen voidaan estää jo GML-tiedoston luontivaiheessa antamalla ogr2ogr-ohjelmalle parametri -dsco XSISHEMA=OFF. Täydellinen toimiva komento olisi siis muotoa

```
ogr2ogr -f GML -dsco XSISHEMA=OFF testi.gml m_meri.tab --config  
OGR_FORCE_ASCII NO
```

- Edellinen komento luo GDAL:in omaa muotoa olevan skeematiedoston, jonka tiedostonimen tarkenne on .gfs
- Jos tehdään peräkkäisiä kokeiluja samoilla tiedostoilla, niin edellisissä muunnoksissa syntyneet .xsd- ja .gfs- tiedostot voivat vaikuttaa tuloksiin. Niiden olemassaolo on muistettava ottaa huomioon, ja yleensä ne on parasta tuhota kokeilujen välillä.

### **Vaihtoehto merkistökoodauksen muuntamiseen: iconv-ohjelma**

GDAL:in puolittain toteutetun GML-tiedostomuodon merkistötuen takia muunnos WindowsLatin1:stä ISO-8859-1:stä UTF-8-koodaukseen on konstikasta. Harkitseminen arvoinen vaihtoehto on tehdä merkistön muunnos nimenomaan siihen tarkoitukseen tehdyllä ohjelmalla kuten esimerkiksi iconv-ohjelmalla. Iconv tukee kymmeniä erilaisia merkistökoodauksia ja Windows-käyttäjä saa sen vaivattomasti käyttöönsä lataamalla GDAL asennusohjelman Gisinternals.com -sivustolta <http://www.gisinternals.com/sdk/>

Koska iconv muuntaa merkkejä, niin Mapinfon TAB-muotoisten tiedostojen koodausta ei voi muuntaa sillä suoraan, vaan välissä on käytettävä jotain sellaista tiedostomuotoa, jossa ominaisuustietojen nimet ja varsinainen tietosisältö ovat tekstimuodossa. GML, jota tässä ohjeessa on käsitelty, on nimenomaan sellainen tiedostomuoto. Toinen vaihtoehto on käyttää Mapinfon tekstipohjaista MID/MIF -tiedostomuotoa. Ogr2ogr osaa tehdä muunnoksen TAB-muodosta MID/MIF -muotoon. Tässä annetaan kuitenkin ohjeet GML-tiedostomuotoa varten.

Seuraavilla komennoilla muunnetaan iconv-ohjelmalla ogr2ogr:n luomat gml- ja xsd-tiedostot UTF-8-koodatuiksi

```
iconv -f ISO-8859-1 -t UTF-8 testi.gml >testi_utf8.gml  
iconv -f ISO-8859-1 -t UTF-8 testi.xsd >testi_utf8.xsd
```

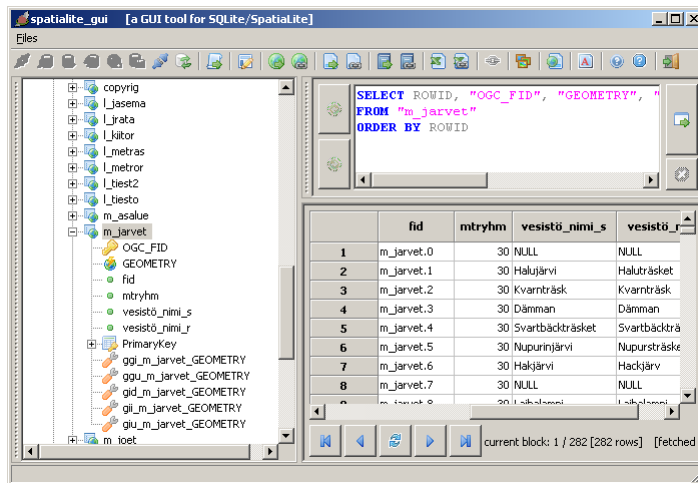
Iconv-ohjelma ei tee mitään muita muutoksia muunnettaviin tiedostoihin eikä esimerkiksi päivitä XML-muotoisten tiedostojen otsikkoriviä. Tässä nimenomaisessa tapauksessa ogr2ogr-ohjelman luomissa tiedostoissa oli valmiiksi sopivalla tavalla väärät otsikot, jotka väittivät sisällön olevan UTF-8-koodattua, vaikka se ei ollutkaan. Iconv-muunnoksen jälkeen otsikkorivi pitää paikkansa, joten sitä ei tarvitse erikseen muokata.

## Tehokäyttäjät tekee komentojonoja

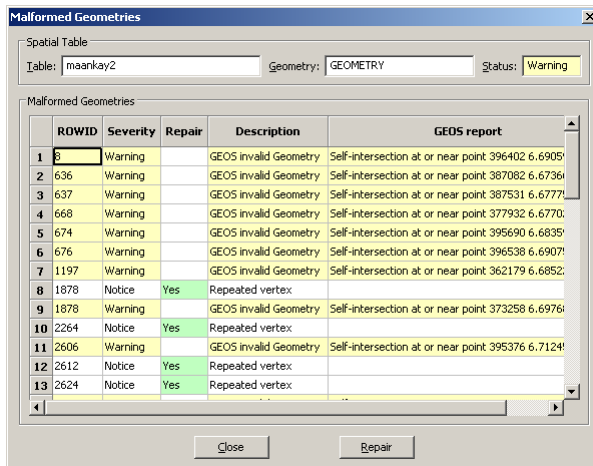
Vaikka tämä ohje onkin pitkä, niin Mapinfo-tiedostojen muuntaminen ogr2ogr- ja iconv-ohjelmilla Spatialite-muotoon on hyvin nopeaa sitten, kun tietää mitä on tekemässä. Seuraavassa on tulokset kaikkien seutukartta-aineistoon kuuluvien 60 karttatason muuntamiseen kuluneesta ajasta vaiheittain. Muunnos tehtiin vuoden 2004 mallia olevalla yksiproessorisella Windows-koneella ja sekä lähtö- että tulosaineistot olivat samalla ulkoisella USB-levyllä. Käytetyt komentojonot ovat tämän ohjeen liitteenä, ja kokeen voi helposti toistaa niiden avulla.

<u>Vaihe</u>	<u>komentojono</u>	<u>kulunut aika</u>
TAB - GML	tee_gml.bat	25 sekuntia
ISO-8859-1 -> UTF-8	aja_iconv.bat	12 sekuntia
GML -> Spatialite	gml_sqlite.bat	65 sekuntia
yhteensä		102 sekuntia

Oheisesta kuvaruutukaappauksesta näkyy, että ööt ja äät näkyvät kuten pitääkin sekä ominaisuustietojen nimissä että niiden arvoissa.

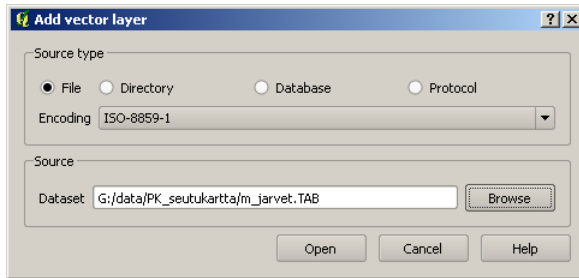


Toisesta kuvaruutukaappauksesta nähdään, että teksisesti onnistunut muunnos tiedostomuodosta toiseen ei korjaa alkuperäisessä aineistossa mahdollisesti olleita topologiavirheitä.

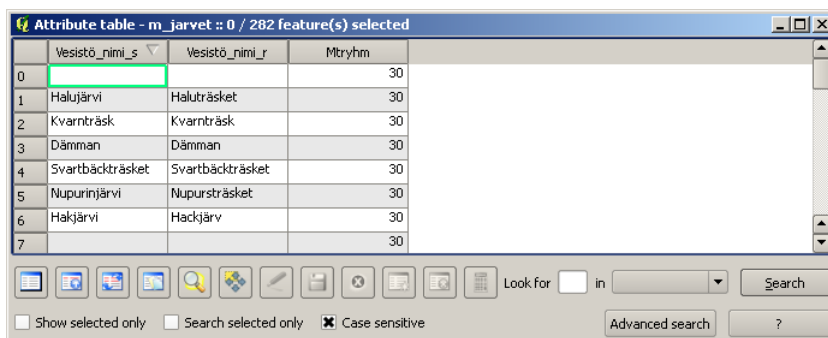


## Satunnainen käyttäjä käyttää Quantum GIS:iä **Ja saattaa kärsiä projektiovirheistä!**

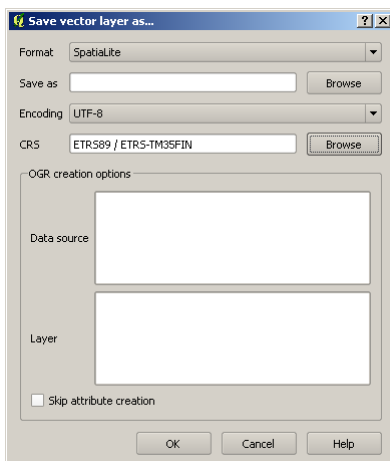
Yksittäisen seutukartta-aineiston tason muuntaminen onnistuu helposti ja suoraviivaisesti Quantum GIS-ohjelmalla. Se avaa Mapinfo-tiedostot suoraan tiedostonavaustoiminnolla ja antaa käyttäjän valita tiedostossa käytetyn merkistökoodauksen suoraan valikosta.



Ominaisuustietojen nimet ja arvot näkyvät tuonnin jälkeen oikein



Tallennusvalikosta voidaan valita tallennusmuoto, koordinaattijärjestelmä ja merkistökoodaus. Koordinaattijärjestelmän vaihtamisessa on kuitenkin huomattava, että vain KKJ-kaistojen välillä tehtävä muunnos menee varmasti oikein. Jos Quantum GIS käyttää vanhempaa GDAL versiota kuin 1.9.0, niin esimerkiksi muunnokset WGS84-järjestelmään tai ETRS-TM35FIN:iin (EPSG:4326 ja EPSG:3067) menevät varmasti väärin. Tätä aihetta käsittelee [Tee muunnokset KKJ->TM35FIN oikein](#). **Alla oleva toimenpide voi siis tuottaa väärän tuloksen.**







```

a_va_kos.TAB --config OGR_FORCE_ASCII no
ogr2ogr -f GML -dsco XSCHEMA=OFF -t_srs epsg:3067 a_va_suu.GML.temp -dsco XSCHEMA=OFF
a_va_suu.TAB --config OGR_FORCE_ASCII no
ogr2ogr -f GML -dsco XSCHEMA=OFF -t_srs epsg:3067 a_vi_pie.GML.temp -dsco XSCHEMA=OFF
a_vi_pie.TAB --config OGR_FORCE_ASCII no
ogr2ogr -f GML -dsco XSCHEMA=OFF -t_srs epsg:3067 a_vi_suu.GML.temp -dsco XSCHEMA=OFF
a_vi_suu.TAB --config OGR_FORCE_ASCII no
ogr2ogr -f GML -dsco XSCHEMA=OFF -t_srs epsg:3067 a_vi_til.GML.temp -dsco XSCHEMA=OFF
a_vi_til.TAB --config OGR_FORCE_ASCII no
ogr2ogr -f GML -dsco XSCHEMA=OFF -t_srs epsg:3067 Copyrig.GML.temp -dsco XSCHEMA=OFF
Copyrig.TAB --config OGR_FORCE_ASCII no
ogr2ogr -f GML -dsco XSCHEMA=OFF -t_srs epsg:3067 l_jasema.GML.temp -dsco XSCHEMA=OFF
l_jasema.TAB --config OGR_FORCE_ASCII no
ogr2ogr -f GML -dsco XSCHEMA=OFF -t_srs epsg:3067 l_jrata.GML.temp -dsco XSCHEMA=OFF
l_jrata.TAB --config OGR_FORCE_ASCII no
ogr2ogr -f GML -dsco XSCHEMA=OFF -t_srs epsg:3067 l_kiitor.GML.temp -dsco XSCHEMA=OFF
l_kiitor.TAB --config OGR_FORCE_ASCII no
ogr2ogr -f GML -dsco XSCHEMA=OFF -t_srs epsg:3067 l_metras.GML.temp -dsco XSCHEMA=OFF
l_metras.TAB --config OGR_FORCE_ASCII no
ogr2ogr -f GML -dsco XSCHEMA=OFF -t_srs epsg:3067 l_metror.GML.temp -dsco XSCHEMA=OFF
l_metror.TAB --config OGR_FORCE_ASCII no
ogr2ogr -f GML -dsco XSCHEMA=OFF -t_srs epsg:3067 l_tiest2.GML.temp -dsco XSCHEMA=OFF
l_tiest2.TAB --config OGR_FORCE_ASCII no
ogr2ogr -f GML -dsco XSCHEMA=OFF -t_srs epsg:3067 l_tiesto.GML.temp -dsco XSCHEMA=OFF
l_tiesto.TAB --config OGR_FORCE_ASCII no
ogr2ogr -f GML -dsco XSCHEMA=OFF -t_srs epsg:3067 Maankay2.GML.temp -dsco XSCHEMA=OFF
Maankay2.TAB --config OGR_FORCE_ASCII no
ogr2ogr -f GML -dsco XSCHEMA=OFF -t_srs epsg:3067 m_asalue.GML.temp -dsco XSCHEMA=OFF
m_asalue.TAB --config OGR_FORCE_ASCII no
ogr2ogr -f GML -dsco XSCHEMA=OFF -t_srs epsg:3067 m_jarvet.GML.temp -dsco XSCHEMA=OFF
m_jarvet.TAB --config OGR_FORCE_ASCII no
ogr2ogr -f GML -dsco XSCHEMA=OFF -t_srs epsg:3067 m_joet.GML.temp -dsco XSCHEMA=OFF
m_joet.TAB --config OGR_FORCE_ASCII no
ogr2ogr -f GML -dsco XSCHEMA=OFF -t_srs epsg:3067 m_meri.GML.temp -dsco XSCHEMA=OFF
m_meri.TAB --config OGR_FORCE_ASCII no
ogr2ogr -f GML -dsco XSCHEMA=OFF -t_srs epsg:3067 m_rantav.GML.temp -dsco XSCHEMA=OFF
m_rantav.TAB --config OGR_FORCE_ASCII no
ogr2ogr -f GML -dsco XSCHEMA=OFF -t_srs epsg:3067 m_teolal.GML.temp -dsco XSCHEMA=OFF
m_teolal.TAB --config OGR_FORCE_ASCII no
ogr2ogr -f GML -dsco XSCHEMA=OFF -t_srs epsg:3067 m_vihral.GML.temp -dsco XSCHEMA=OFF
m_vihral.TAB --config OGR_FORCE_ASCII no
ogr2ogr -f GML -dsco XSCHEMA=OFF -t_srs epsg:3067 N_KOS_R.GML.temp -dsco XSCHEMA=OFF
N_KOS_R.TAB --config OGR_FORCE_ASCII no
ogr2ogr -f GML -dsco XSCHEMA=OFF -t_srs epsg:3067 N_KOS_S.GML.temp -dsco XSCHEMA=OFF
N_KOS_S.TAB --config OGR_FORCE_ASCII no
ogr2ogr -f GML -dsco XSCHEMA=OFF -t_srs epsg:3067 N_MERI_R.GML.temp -dsco XSCHEMA=OFF
N_MERI_R.TAB --config OGR_FORCE_ASCII no
ogr2ogr -f GML -dsco XSCHEMA=OFF -t_srs epsg:3067 N_MERI_S.GML.temp -dsco XSCHEMA=OFF
N_MERI_S.TAB --config OGR_FORCE_ASCII no
ogr2ogr -f GML -dsco XSCHEMA=OFF -t_srs epsg:3067 N_PAIK_R.GML.temp -dsco XSCHEMA=OFF
N_PAIK_R.TAB --config OGR_FORCE_ASCII no
ogr2ogr -f GML -dsco XSCHEMA=OFF -t_srs epsg:3067 N_PAIK_S.GML.temp -dsco XSCHEMA=OFF
N_PAIK_S.TAB --config OGR_FORCE_ASCII no
ECHO ON
time

```

## aja\_iconv.bat

```
=====
```

```

time
@ECHO OFF
iconv -f ISO-8859-1 -t UTF-8 A_es_pie.GML.temp >A_es_pie.GML
iconv -f ISO-8859-1 -t UTF-8 A_es_suu.GML.temp >A_es_suu.GML
iconv -f ISO-8859-1 -t UTF-8 a_es_til.GML.temp >a_es_til.GML
iconv -f ISO-8859-1 -t UTF-8 a_hk_osa.GML.temp >a_hk_osa.GML
iconv -f ISO-8859-1 -t UTF-8 a_hk_per.GML.temp >a_hk_per.GML
iconv -f ISO-8859-1 -t UTF-8 a_hk_pie.GML.temp >a_hk_pie.GML
iconv -f ISO-8859-1 -t UTF-8 a_hk_suu.GML.temp >a_hk_suu.GML
iconv -f ISO-8859-1 -t UTF-8 a_hy_pie.GML.temp >a_hy_pie.GML
iconv -f ISO-8859-1 -t UTF-8 a_hy_suu.GML.temp >a_hy_suu.GML
iconv -f ISO-8859-1 -t UTF-8 a_hy_til.GML.temp >a_hy_til.GML
iconv -f ISO-8859-1 -t UTF-8 a_ja_pie.GML.temp >a_ja_pie.GML
iconv -f ISO-8859-1 -t UTF-8 a_ja_til.GML.temp >a_ja_til.GML
iconv -f ISO-8859-1 -t UTF-8 a_ka_pie.GML.temp >a_ka_pie.GML
iconv -f ISO-8859-1 -t UTF-8 a_ka_til.GML.temp >a_ka_til.GML
iconv -f ISO-8859-1 -t UTF-8 a_ke_pie.GML.temp >a_ke_pie.GML
iconv -f ISO-8859-1 -t UTF-8 a_ke_til.GML.temp >a_ke_til.GML
iconv -f ISO-8859-1 -t UTF-8 a_ki_pie.GML.temp >a_ki_pie.GML
iconv -f ISO-8859-1 -t UTF-8 a_ki_suu.GML.temp >a_ki_suu.GML

```

```

iconv -f ISO-8859-1 -t UTF-8 a_ki_til.GML.temp >a_ki_til.GML
iconv -f ISO-8859-1 -t UTF-8 a_kunta.GML.temp >a_kunta.GML
iconv -f ISO-8859-1 -t UTF-8 a_ma_pie.GML.temp >a_ma_pie.GML
iconv -f ISO-8859-1 -t UTF-8 a_ma_til.GML.temp >a_ma_til.GML
iconv -f ISO-8859-1 -t UTF-8 a_nu_pie.GML.temp >a_nu_pie.GML
iconv -f ISO-8859-1 -t UTF-8 a_nu_suu.GML.temp >a_nu_suu.GML
iconv -f ISO-8859-1 -t UTF-8 a_nu_til.GML.temp >a_nu_til.GML
iconv -f ISO-8859-1 -t UTF-8 a_pkspie.GML.temp >a_pkspie.GML
iconv -f ISO-8859-1 -t UTF-8 a_pkstil.GML.temp >a_pkstil.GML
iconv -f ISO-8859-1 -t UTF-8 a_pksuur.GML.temp >a_pksuur.GML
iconv -f ISO-8859-1 -t UTF-8 a_po_til.GML.temp >a_po_til.GML
iconv -f ISO-8859-1 -t UTF-8 a_si_pie.GML.temp >a_si_pie.GML
iconv -f ISO-8859-1 -t UTF-8 a_tu_pie.GML.temp >a_tu_pie.GML
iconv -f ISO-8859-1 -t UTF-8 a_tu_suu.GML.temp >a_tu_suu.GML
iconv -f ISO-8859-1 -t UTF-8 a_tu_til.GML.temp >a_tu_til.GML
iconv -f ISO-8859-1 -t UTF-8 a_va_kos.GML.temp >a_va_kos.GML
iconv -f ISO-8859-1 -t UTF-8 a_va_suu.GML.temp >a_va_suu.GML
iconv -f ISO-8859-1 -t UTF-8 a_vi_pie.GML.temp >a_vi_pie.GML
iconv -f ISO-8859-1 -t UTF-8 a_vi_suu.GML.temp >a_vi_suu.GML
iconv -f ISO-8859-1 -t UTF-8 a_vi_til.GML.temp >a_vi_til.GML
iconv -f ISO-8859-1 -t UTF-8 Copyrig.GML.temp >Copyrig.GML
iconv -f ISO-8859-1 -t UTF-8 l_jasema.GML.temp >l_jasema.GML
iconv -f ISO-8859-1 -t UTF-8 l_jrata.GML.temp >l_jrata.GML
iconv -f ISO-8859-1 -t UTF-8 l_kiitor.GML.temp >l_kiitor.GML
iconv -f ISO-8859-1 -t UTF-8 l_metras.GML.temp >l_metras.GML
iconv -f ISO-8859-1 -t UTF-8 l_metror.GML.temp >l_metror.GML
iconv -f ISO-8859-1 -t UTF-8 l_tiest2.GML.temp >l_tiest2.GML
iconv -f ISO-8859-1 -t UTF-8 l_tiesto.GML.temp >l_tiesto.GML
iconv -f ISO-8859-1 -t UTF-8 Maankay2.GML.temp >Maankay2.GML
iconv -f ISO-8859-1 -t UTF-8 m_asalue.GML.temp >m_asalue.GML
iconv -f ISO-8859-1 -t UTF-8 m_jarvet.GML.temp >m_jarvet.GML
iconv -f ISO-8859-1 -t UTF-8 m_joet.GML.temp >m_joet.GML
iconv -f ISO-8859-1 -t UTF-8 m_meri.GML.temp >m_meri.GML
iconv -f ISO-8859-1 -t UTF-8 m_rantav.GML.temp >m_rantav.GML
iconv -f ISO-8859-1 -t UTF-8 m_teolal.GML.temp >m_teolal.GML
iconv -f ISO-8859-1 -t UTF-8 m_vihral.GML.temp >m_vihral.GML
iconv -f ISO-8859-1 -t UTF-8 N_KOS_R.GML.temp >N_KOS_R.GML
iconv -f ISO-8859-1 -t UTF-8 N_KOS_S.GML.temp >N_KOS_S.GML
iconv -f ISO-8859-1 -t UTF-8 N_MERI_R.GML.temp >N_MERI_R.GML
iconv -f ISO-8859-1 -t UTF-8 N_MERI_S.GML.temp >N_MERI_S.GML
iconv -f ISO-8859-1 -t UTF-8 N_PAIK_R.GML.temp >N_PAIK_R.GML
iconv -f ISO-8859-1 -t UTF-8 N_PAIK_S.GML.temp >N_PAIK_S.GML
ECHO ON
time

```

## gml\_sqlite.bat

=====

```

time
@echo off
ogr2ogr -f SQLite -dsco spatialite=yes -dsco init_with_epsg=yes -gt 2048 -a_srs epsg:3067
pk_seutukartta.sqlite A_es_pie.GML -nlt polygon -lco LAUNDER=yes
ogr2ogr -f SQLite -update -gt 2048 -a_srs epsg:3067 pk_seutukartta.sqlite A_es_suu.GML -nlt polygon
-lco LAUNDER=yes
ogr2ogr -f SQLite -update -gt 2048 -a_srs epsg:3067 pk_seutukartta.sqlite a_es_til.GML -nlt polygon
-lco LAUNDER=yes
ogr2ogr -f SQLite -update -gt 2048 -a_srs epsg:3067 pk_seutukartta.sqlite a_hk_osa.GML -nlt polygon
-lco LAUNDER=yes
ogr2ogr -f SQLite -update -gt 2048 -a_srs epsg:3067 pk_seutukartta.sqlite a_hk_per.GML -nlt polygon
-lco LAUNDER=yes
ogr2ogr -f SQLite -update -gt 2048 -a_srs epsg:3067 pk_seutukartta.sqlite a_hk_pie.GML -nlt polygon
-lco LAUNDER=yes
ogr2ogr -f SQLite -update -gt 2048 -a_srs epsg:3067 pk_seutukartta.sqlite a_hk_suu.GML -nlt polygon
-lco LAUNDER=yes
ogr2ogr -f SQLite -update -gt 2048 -a_srs epsg:3067 pk_seutukartta.sqlite a_hy_pie.GML -nlt polygon
-lco LAUNDER=yes
ogr2ogr -f SQLite -update -gt 2048 -a_srs epsg:3067 pk_seutukartta.sqlite a_hy_suu.GML -nlt polygon
-lco LAUNDER=yes
ogr2ogr -f SQLite -update -gt 2048 -a_srs epsg:3067 pk_seutukartta.sqlite a_hy_til.GML -nlt polygon
-lco LAUNDER=yes
ogr2ogr -f SQLite -update -gt 2048 -a_srs epsg:3067 pk_seutukartta.sqlite a_ja_pie.GML -nlt polygon
-lco LAUNDER=yes
ogr2ogr -f SQLite -update -gt 2048 -a_srs epsg:3067 pk_seutukartta.sqlite a_ja_til.GML -nlt polygon
-lco LAUNDER=yes
ogr2ogr -f SQLite -update -gt 2048 -a_srs epsg:3067 pk_seutukartta.sqlite a_ka_pie.GML -nlt polygon
-lco LAUNDER=yes
ogr2ogr -f SQLite -update -gt 2048 -a_srs epsg:3067 pk_seutukartta.sqlite a_ka_til.GML -nlt polygon

```



```
multipolygon -lco LAUNDER=yes
ogr2ogr -f SQLite -update -gt 2048 -a_srs epsg:3067 pk_seutukartta.sqlite N_KOS_R.GML -nlt point
-lco LAUNDER=yes
ogr2ogr -f SQLite -update -gt 2048 -a_srs epsg:3067 pk_seutukartta.sqlite N_KOS_S.GML -nlt point
-lco LAUNDER=yes
ogr2ogr -f SQLite -update -gt 2048 -a_srs epsg:3067 pk_seutukartta.sqlite N_MERI_R.GML -nlt point
-lco LAUNDER=yes
ogr2ogr -f SQLite -update -gt 2048 -a_srs epsg:3067 pk_seutukartta.sqlite N_MERI_S.GML -nlt point
-lco LAUNDER=yes
ogr2ogr -f SQLite -update -gt 2048 -a_srs epsg:3067 pk_seutukartta.sqlite N_PAIK_R.GML -nlt point
-lco LAUNDER=yes
ogr2ogr -f SQLite -update -gt 2048 -a_srs epsg:3067 pk_seutukartta.sqlite N_PAIK_S.GML -nlt point
-lco LAUNDER=yes
ECHO ON
time
```