

GDAL ja WFS: suora muunnos ogr2ogr-ohjelmalla

Viimeksi muokattu 30. tammikuuta 2012

GDAL tukee WFS-palvelun käyttämistä vektoritietolähteenä yhtenä muiden joukossa. WFS-ajurin käyttöohjeet löytyvät osoitteesta http://gdal.org/ogr/drv_wfs.html. Tässä ohjeessa jaetaan taustatietoa, käytännön kokemuksia ja toimivia esimerkkejä aiheesta.

GDAL voi muuntaa vektoritiedot WFS-palvelusta suoraan mihin tahansa niistä tiedosto- ja tietokantamuodoista, joilla on seuraavasta linkistä aukeavassa taulukossa ominaisuus ”Creation – Yes”. http://gdal.org/ogr/ogr_formats.html

Kaikkia luettolon tiedostomuotoja ei kuitenkaan aina tueta. Käytössä olevan GDAL-version tukemat tiedostomuodot voi aina tarkistaa antamalla tämän komennon:

```
ogrinfo --formats
```

Vastaavasti voi tarkistaa käytössä olevan GDAL-version komennolla

```
ogrinfo --version
```

Sopivan GDAL-version asentaminen

WFS-tukeen tarvitaan vähintään GDAL/OGR-versio 1.8.0, mutta hyödylliset GetCapabilities ja LayerMetadata -kyselyt tulivat mukaan vasta versioon 1.9.0. Tämä versio julkaistiin tammikuussa 2012.

Helppoja tapoja GDAL:in asentamiseksi on käyttää Windows-asennuspaketteja sivustolta <http://www.gisinternals.com/sdk/> tai asentaa koneelle OSGeo4W-paketti. Tosin OSGeo4W sisältää vielä tammikuun lopussakin GDAL-version 1.8.0.

WFS ja ogrinfo

Kun GDAL-ohjelmat on asennettu, niin WFS-palvelun käyttäminen GDAL-komentoriviltä on yksinkertaista.

Esimerkki 1.

Lue WFS-palvelun ominaisuudet

```
ogrinfo WFS:http://hip.latuviitta.org/cgi-bin/tinyows WFSGetCapabilities
```

Esimerkki 2.

Jos listaus sattuu olemaan pitkä, kannattaa ominaisuudet lukea ketjuttamalla tuloste Windowsin ”more” ohjelman kautta (pystyviiva saadaan painamalla AltGr ja vasemman vaihtonäppäimen vieressä oleva pienempi kuin/suurempi kuin -näppäintä).

```
ogrinfo WFS:http://hip.latuviitta.org/cgi-bin/tinyows WFSGetCapabilities|more
```

Esimerkki 3.

Lue WFS-palvelun ominaisuudet ja ohjaa tulos tiedostoon ”latuviitta.txt”

```
ogrinfo WFS:http://hip.latuviitta.org/cgi-bin/tinyows WFSGetCapabilities  
>latuviitta.txt
```

Esimerkki 4.

Lue lyhyt yhteenveto kaikista palvelussa olevista WFS-tasoista

```
ogrinfo WFS:http://hip.latuviitta.org/cgi-bin/tinyows WFSLayerMetadata
```

VAROITUS! Käytä seuraavaa komentoa harkiten

WFS-tasojen ominaisuuksien kysely ogrinfo-työkalulla tekee GetFeature-kyselyn WFS-palvelusta eli se hakee kyseltävän tason kaikkien kohteiden geometriat ja ominaisuustiedot. Jos taso on suuri, niin haku voi olla hidaskäyttöinen. Jos aikomuksena on ladata taso joka tapauksessa myöhemmin, on parempi ladata taso ensin ja kysellä sen ominaisuuksia sitten paikallisesti tallennetusta tiedostosta. Tällä tavalla ei tehdä kuin yksi GetFeature-pyyntö palvelusta.

Esimerkki 5.

Lue yhteenvetotiedot WFS-tasosta ”lv_pks_tilastoalue_piste”

```
ogrinfo -so WFS:http://hip.latuviitta.org/cgi-bin/tinyows  
lv:pks_tilastoalue_piste
```

Huomaa parametri -so ”summary only”. Ilman sitä komento listaa myös jokaisen kohteen. Sillä ei kuitenkaan ole vaikutusta ogrinfon tekemään kyselyyn, palvelimelta haetaan joka tapauksessa kaikki tason kohteet.

Mitä ogrinfo juttelee WFS-palvelimen kanssa?

Otetaan esimerkiksi edellisen esimerkin 5. ogrinfo-pyyntö

```
ogrinfo -so WFS:http://hip.latuviitta.org/cgi-bin/tinyows  
lv:pks_tilastoalue_piste
```

Komennon jälkeen GDAL lähettää yhteensä neljä pyyntöä WFS-palveluun. Näistä ensimmäinen on normaali GetCapabilities-pyyntö

[http://hip.latuviitta.org/cgi-bin/tinyows?
SERVICE=WFS&REQUEST=GetCapabilities&ACCEPTVERSIONS=1.1.0,1.0.0](http://hip.latuviitta.org/cgi-bin/tinyows?SERVICE=WFS&REQUEST=GetCapabilities&ACCEPTVERSIONS=1.1.0,1.0.0)

Seuraavaksi GDAL selvittää WFS-tason rakenteen. Vaikka ogrinfoa kysyttiin vain pks_tilastoalue_piste -tasolta, niin GDAL lähettää saman tien DescribeFeatureType-pyyntöä kaikille muillekin samana nimiavaruutta (lv:) käyttäville WFS-tasoille.

```
http://hip.latuviitta.org/cgi-bin/tinyows?  
SERVICE=WFS&VERSION=1.1.0&REQUEST=DescribeFeatureType&TYPENAME=lv:pks_tilastoalue_piste,lv:mm1
```

_kunta10k_2011_l,lv:mm1_kunta10k_2011_p,lv:mm1_kunta100k_2011_l,lv:mm1_kunta100k_2011_p,lv:mm1_kunta250k_2011_l,lv:mm1_kunta250k_2011_p,lv:mm1_kunta1000k_2011_l,lv:mm1_kunta1000k_2011_p,lv:mm1_kunta4500k_2011_l,lv:mm1_kunta4500k_2011_p,lv:municipalities,lv:mm1_airport,lv:mm1_asetat,lv:mm1_avi1_l,lv:mm1_avi1_p,lv:mm1_cityp,lv:mm1_coast_l,lv:mm1_coast_p,lv:mm1_dcont_l,lv:mm1_dcont_p,lv:mm1_forest,lv:mm1_hcont_l,lv:mm1_hcont_p,lv:mm1_hpoint,lv:mm1_kunta1_l,lv:mm1_kunta1_p,lv:mm1_kunta1_maa_alue,lv:mm1_lake_l,lv:mm1_lake_p,lv:mm1_maaku1_l,lv:mm1_maaku1_p,lv:mm1_namep,lv:mm1_pelto,lv:mm1_railway,lv:mm1_river,lv:mm1_rivera_l,lv:mm1_rivera_p,lv:mm1_road,lv:mm1_suot,lv:mm1_taaajama,lv:mm1_paikannimet20,lv:peltolohkot_2011,lv:pks_pienalue,lv:pks_suuralue,lv:pks_tilastoalue,lv:pks_pienalue_piste,lv:pks_suuralue_piste,lv:natura2000-suomi_2010

Kolmannella pyynnöllä GDAL selvittää, kuinka monta kohdetta WFS-tasolla on lähettämällä GetFeature-pyyntöä ja käyttämällä WFS 1.1.0 -standardin RESULTTYPE=hits -parametria.

[http://hip.latuviitta.org/cgi-bin/tinyows?
SERVICE=WFS&VERSION=1.1.0&REQUEST=GetFeature&TYPENAME=lv:pks_tilastoalue_piste&RESULTTYPE=hits](http://hip.latuviitta.org/cgi-bin/tinyows?SERVICE=WFS&VERSION=1.1.0&REQUEST=GetFeature&TYPENAME=lv:pks_tilastoalue_piste&RESULTTYPE=hits)

Neljäs pyyntö, jonka GDAL lähettää on GetFeature pyyntö ilman mitään muita parametreja.

[http://hip.latuviitta.org/cgi-bin/tinyows?
SERVICE=WFS&VERSION=1.1.0&REQUEST=GetFeature&TYPENAME=lv:pks_tilastoalue_piste](http://hip.latuviitta.org/cgi-bin/tinyows?SERVICE=WFS&VERSION=1.1.0&REQUEST=GetFeature&TYPENAME=lv:pks_tilastoalue_piste)

Mitä taustalla tapahtuu?

Ogrinfo-ohjelman lähettämät pyynnot ovat periaatteessa järkeviä, mutta niistä myös huomaa, että WFS-kehittäjät kehittävät ohjelmiaan epärealistisen pieniä WFS-tasoja vasten.

1. GetCapabilities

Tämä on normaali aloitus, kun ruvetaan keskustelemaan WFS-palvelun kanssa. Vastauksen perusteella GDAL saa tietää, että tilastoaluepisteet käyttävät EPSG:3067-koordinaattijärjestelmää. Muuten tätä tietoa ei saakaan, paitsi tekemällä GetFeature-kyselyn ja katsomalla, mitä palvelusta tulee ulos.

2. DescribeFeatureType

Tämä on normaali toinen pyyntö WFS-palvelimelle ja sen avulla saadaan selville WFS-tason rakenne eli skeema. GDAL lähettää yhdellä kertaa pyynnön useiden WFS-tasojen skeeman lukemiseksi koska monet WFS-palvelimet vastaavat tällä tavalla nopeammin kuin jos tietoja pyydetäisiin joka tasolta erillisillä pyynnöillä. Vaikka tässä tapauksessa kiinnostaakin vain yhden WFS-tason skeema, niin ei useamman tason tietojen kyselemisestä ei ole suurta haittaa.

3. GetFeature, resulttype=hits

WFS 1.1.0 -version tukema resulttype=hits palauttaa WFS-pyyntöä löytämien kohteiden lukumäärän. Tämä on hyvä tapa, se rasittaa niin vähän kuin mahdollista WFS-palvelinta ja vastaus tulee nopeasti.

4. GetFeature

Pelkkä GetFeature hakee WFS-palvelusta kaikki kohteet, mitä tasolta löytyy, tai sitten niin monta kohdetta kuin palvelimen puolella asetettu maxFeatures-rajoitus sallii. Ogrinfo haluaa siis hakea WFS-palvelimelta kaikki vektorit ja ominaisuustiedot, mutta miksi ihmeessä?

GDAL/ogr on tehty alun perin paikallisten vektoriaineistojen käsittelemiseen ja WFS-ajuri on tehty toimimaan samalla tavalla kuin muutkin ajurit. GDAL haluaa lukea kaikki kohteet, koska se haluaa selvittää luotettavasti aineiston maantieteelliset ulottuvuudet. Kaikilla vektoritietolähteillä ei ole mitään keinoa kertoa näitä ulottuvuuksia, ja monet, jotka siihen pystyvät, saattavat kuitenkin käytännössä ilmoittaa ne väärin. Siitä syystä GDAL:n tapana

on lukea kaikki kohteet ja selvittää todellisten geometrioiden perusteella aineiston ulottuvuudet. Tulos on luotettava ja paikallisessa yhden käyttäjän ympäristössä menettely on ihan järkevä. Sen sijaan WFS-käytössä tässä ei välttämättä ole paljonkaan järkeä. Tasolla voi olla miljoonia kohteita ja niiden kaikkien lukeminen voi merkitä helppostikin satojen megatavujen suuruista GML-muotoista tiedonsiirtoa, mikä on kova hinta pelkkien ulottuvuuksien selvittämiseksi, varsinkin kun WFS-palvelu ilmoittaa tasojen ulottuvuudet jo GetCapabilities-vastauksessa. Ei tosin ole epätavallista, että palvelu ilmoittaa ne väärin, mutta myöskään GDAL:in menetelmä ei tuota aina luotettavaa vastausta. Jos nimittäin palvelimelle on asetettu maxFeatures-rajoitus, niin GDAL laskee ulottuvuudet palvelun lähettämästä vektorijoukosta ja väittää, että ne pätevät koko WFS-tasolle. Jos kohteet on palvelussa järjestetty sijainnin perusteella, niin tason alkupään kohteiden mukaan lasketut ulottuvuudet voivat olla paljon pienemmät kuin koko tason ulottuvuudet.

Erityisen tappavaa on antaa seuraava komento:

```
ogrinfo -al WFS:http://hip.latuviitta.org/cgi-bin/tinyows
```

Tällä johtaa siihen, että ogrinfo lähettää WFS-palvelun jokaiselle aineistolle vuoron perään kaksi GetFeature-pyyntöä, ensimmäinen parametrilla resulttype=hits ja toisen ilman tätä. Esimerkkipalvelussa on yli 50 WFS-tasoa, joista useisiin sisältyy satoja tuhansia kohteita. Jos pyyntö menisi läpi, niin sen käsittelemiseen menisi aikaa useita tunteja ja tuloksena olisi muutama gigatavu verkkoliikennettä.

Kohteiden hakeminen WFS:stä ogr2ogr-ohjelmalla

Esimerkki 6.

Hae WFS-taso pks_tilastoalue_piste shapefileksi ja käytä palvelun oletuskoordinaattijärjestelmää

```
ogr2ogr -f "ESRI Shapefile" points.shp WFS:http://hip.latuviitta.org/cgi-bin/tinyows lv:pks_tilastoalue_piste
```

Esimerkki 7.

Sama juttu, mutta muunna saman tien shapefile EPSG:4326 -järjestelmään

```
ogr2ogr -f "ESRI Shapefile" -t_srs epsg:4326 points.shp WFS:http://hip.latuviitta.org/cgi-bin/tinyows lv:pks_tilastoalue_piste
```

Esimerkki 8.

Muunnetaan EPSG:4326 -järjestelmään, mutta tällä kertaa MapInfo:n tiedostomuotoon (TAB).

```
ogr2ogr -f "MapInfo file" -t_srs epsg:4326 points.tab WFS:http://hip.latuviitta.org/cgi-bin/tinyows lv:pks_tilastoalue_piste
```

Esimerkki 9.

Haetaan tiedot uuteen Spatialite-tietokantaan, annetaan uudelle tietokantataululle nimeksi ”tilastoaluepisteet” ja annetaan varmuuden vuoksi ohje, että tasosta tehdään nimenomaan pisteitä sisältävä taso.

```
ogr2ogr -f SQLite -dsco spatialite=yes -dsco init_with_epsg=yes -nln
tilastoaluepisteet -nlt point uusi_spatialite.sqlite
WFS:http://hip.latuviitta.org/cgi-bin/tinyows lv:pk_s_tilastoalue_piste
```

Huom. Komento tuottaa virheen, koska ogr2ogr haluaisi luoda pääavainkentän ogc_fid, mutta saman niminen kenttä on jo lähtöaineistossa. Mielenkiintoinen tilanne, ja siksi jätetty jäljelle tähän ohjeeseen, vaikka kätevää ratkaisua ei olekaan tiedossa. Luultavasti -sql -parametrin käyttäminen olisi yksi vaihtoehto.

Esimerkki 10.

Leikkaa WFS-tasosta pala, jonka ulottuvuudet ovat 377346 6673628 392155 6684806 ja tallenna ne KKJ 2 -kaistaan. Huomaa, että ulottuvuudet on annettava WFS-tason oletusjärjestelmässä, joka tässä tapauksessa on EPSG:3067, ja jonka saa selville GetCapabilities-vastauksesta.

```
ogr2ogr -f "ESRI Shapefile" -spat 377346 6673628 392155 6684806 -t_srs epsg:2392
points2.shp WFS:http://hip.latuviitta.org/cgi-bin/tinyows
lv:pk_s_tilastoalue_piste
```

Esimerkki 11.

Ogrinfolla voi tehdä yllättäviä asioita.

```
ogrinfo WFS:http://hip.latuviitta.org/cgi-bin/tinyows -sql "select kunta,avi
from municipalities"
```