

GUÍA PARA USAR DIFERENTES FORMATOS DE DATOS VECTOR CON MAPSERVER

INFORMACIÓN VECTOR (VECTOR DATA)

Vector: “Es una abstracción del mundo real en donde la información de posición / localización es representada en forma de coordenadas. En la información vector, las unidades básicas de información espacial son los puntos, las líneas y los polígonos. Cada una de estas unidades esta conformada por una serie de uno o más puntos coordenados. Por ejemplo, una línea es una colección de puntos relacionados, y un polígono es una colección de líneas relacionadas. Las imágenes vector son definidas matemáticamente como una serie de puntos conectados por líneas.”

Traducción de http://coris.noaa.gov/glossary/glossary_l_z.html#v

MAPSERVER Y EL ACCESO A LA INFORMACIÓN EN DIFERENTES FORMATOS VECTOR

MapServer tiene dos métodos principales para acceder la información:

A través de propiedades preconfiguradas de acceso a formatos específicos. La forma más sencilla de acceder a información en diferentes formatos es usar las propiedades incorporadas a MapServer cuando este fue compilado. Estas propiedades se limitan a algunos formatos de información, como el ESRI Shapefile, PostGIS, Oracle Spatial y ArcSDE. El formato incorporado siempre por defecto a MapServer es el ESRI Shapefile.

A través del uso de nuevas fuentes de librerías de acceso a datos. Uno de los atributos más poderosos con que cuenta MapServer es su habilidad de leer nuevos formatos de información a través de librerías de acceso a datos. La mas comúnmente usada es la GDAL/OGR, con la cual MapServer accede a diferentes formatos raster (GDAL) y vector (OGR).

Uso de OGR

La librería OGR es usada cuando MapServer lo requiere. MapServer no necesita OGR para funcionar. Algunos usuarios pueden no necesitar las características que ofrece OGR, pero otros pueden considerarlas absolutamente necesarias. Debido a que MapServer puede acceder a diferentes formatos de información por medio de OGR, hay una necesidad menor de programar MapServer para que soporte formatos específicos.

TIPOS DE FORMATO DE DATOS (DATA FORMAT TYPES)

Existe tres tipos de formatos de datos de mapeo y SIG. Cada uno se maneja diferente. A continuación se enuncian los tipos y se dan algunos ejemplos de formatos.

Basados en archivos (File-based) - Shapefiles, Microstation Design Files (DGN), imágenes GeoTIFF.

Basados en directorios (Directory-based) - Coberturas ESRI ArcInfo, US Census TIGRE

Conexiones a Bases de Datos (Database connections) - PostGIS, ESRI ArcSDE, MySQL

Cada tipo de dato tiene una fuente de datos (data source) y una o más capas de información (layers). Las siguientes definiciones son aplicables a MapServer

Fuente de Datos - Es un grupo de capas de información almacenados en un repositorio de datos común. Este puede ser un archivo que maneja múltiples capas de información, o un directorio que contiene múltiples archivos.

Capa de Información - Es un sub-conjunto de la Fuente de Datos que contiene información de un solo tipo (puntos, líneas, polígonos).

Datos Basados en archivos (File-based Data)

Los datos basados en archivo se componen de uno o más archivos almacenados en un directorio arbitrario. En muchos casos solo existe un archivo, pero los ESRI Shapefiles, por ejemplo, se componen de al menos tres archivos con extensiones distintas: .shp, .shx, .dbf. En este caso se requieren los tres archivos ya que cada uno cumple con una tarea diferente. Los nombres de los archivos son utilizados comúnmente como el nombre de la fuente de datos.

Datos Basados en directorios (Directory-based Data)

Los datos basados en directorios se componen de uno o más archivos almacenados de manera particular en un directorio principal. En algunos casos, las coberturas por ejemplo, también se requieren directorios adicionales en otras ubicaciones para que se pueda acceder a la información. El directorio mismo puede ser la fuente de datos. Los diferentes archivos dentro de un directorio a menudo representan las capas de información disponibles. Por ejemplo las Coberturas ESRI ArcInfo se componen de más de un archivo con la extensión .adf dentro de un mismo directorio. El archivo pal.adf representa los polígonos, arc.adf contiene la información de arcos o líneas. El directorio contiene la fuente de datos y cada archivo .adf es una capa de información.

Conexiones a Bases de Datos (Database connections)

Las conexiones a Bases de Datos son muy similares a las estructuras de los datos basados en archivos y datos basados en directorios en un aspecto: proveen a MapServer información de coordenadas geográficas para que esta la interprete. Esto puede ser una simplificación exagerada de lo que ocurre al interior de MapServer, pero en esencia solo se requiere el acceso a las coordenadas que conforman el conjunto de datos vector (vector dataset).

Las conexiones a Bases de Datos proveen una corriente de información coordinada que es almacenada temporalmente (en memoria) y es leída por MapServer para crear el mapa. También puede requerirse otra información tabular o de atributos, pero la principal es la información coordinada. En este punto debe hacerse una importante distinción entre Bases de Datos. Las Bases de Datos que se discuten en ese documento son Bases de Datos Espaciales, aquellas que pueden contener información geográfica

entre sus tipos de datos. Estas son distintas a las Bases de Datos estrictamente tabulares que no pueden contener información geográfica. Existen extensiones espaciales para muchas Bases de Datos (comerciales y de tipo Open Source). Una de las más robustas es la extensión PostGIS para la Base de Datos PostgreSQL. Esta base de datos no solo permite el almacenamiento de información geográfica, también permite la manipulación de esa información por medio de comandos SQL. Otra Base de Datos Open Source con capacidad de incorporar información espacial es MySQL

Las conexiones a las Bases de Datos usualmente se componen de las siguientes piezas de información de conexión:

Host Dirección del servidor o equipo que contiene la Base de Datos.

Database name El nombre de la Base de Datos a la cual se quiere acceder y se encuentra en el host.

User name / passwords Usualmente existen restricciones de acceso a los usuarios. Algunas Bases de Datos, (ej Oracle) utilizan un identificador de servicio (name service identifier) que incluye el nombre del host y de la Base de Datos.

Table/View name Nombre de la tabla o la vista que contiene la información coordenada.

Geographic column name Columna de la tabla en la cual se almacena la geometría o las coordenadas.

GUÍA DE FORMATOS (DATA FORMAT GUIDE)

El resto del documento es la guía de formatos. Estructurada para mostrar los aspectos fundamentales de cada formato de datos soportado por MapServer. Se muestra un resumen con la información más importante de cada formato, seguida de un ejemplo de sintaxis de un mapfile de MapServer.

ESRI Shapefiles (SHP)

También conocidos como ArcView ESRI Shapefiles o ESRI Shapefiles. ESRI es la compañía que introdujo este formato.

Los Shapefiles se componen de un mínimo de tres archivos con nombres similares y extensiones distintas, ej:

- Countries_area.dbf
- Countries_area.shp
- Countries_area.shx

Acceso a la información

El acceso a Shapefiles es un atributo que tiene por defecto el MapServer. También se puede acceder a estos a través de OGR, pero se recomienda el acceso directo. Se requiere la ruta al Shapefile, no se debe especificar ninguna extensión. Los Shapefiles solo contienen una capa de información, por lo tanto para distinguir entre una capa y otra solo se necesita su nombre.

Ejemplo de Mapfile

```
LAYER
  NAME my_shapefile
  TYPE POLYGON
  DATA countries_area
  STATUS OFF
  CLASS
    NAME "Countries"
    OUTLINECOLOR 0 0 0
  END
END
```

PostGIS / PostgreSQL Database

PostGIS es una extensión de Refraction Research's para la Base de Datos PostgreSQL.

Soporte PostGIS

El acceso a Bases de Datos PostGIS / PostgreSQL es un atributo que no viene por defecto con MapServer, debe ser compilado con MapServer para que funcione. En la mayoría de los casos las librerías PostgreSQL y PostGIS (.dll o .so) deben estar presentes en los directorios de sistema de MapServer (ej: librerías libpq y libpostgis en el directorio cgi-bin del servidor Apache).

Ejemplo de Mapfile

Se debe incluir la línea de código CONNECTIONTYPE POSTGIS. Se debe definir CONNECTION así:

```
host=nombre_del_host
dbname=nombre_base_datos
user=nombre_usuario_base_datos
password=contraseña_base_datos
port=puerto_pg
```

Los parámetros incluidos en CONNECTION pueden estar en cualquier orden. La mayoría son opcionales. dbname es requerido (obligatorio), host tiene como valor por defecto localhost, port tiene como valor por defecto 5432 - el puerto estándar de PostgreSQL.

Se debe definir DATA así: "columna_geometria de nombre_tabla_datos". Se recomienda usar letras minúsculas para evitar problemas. El parámetro columna_geometría puede tomar el valor the_geom si se utiliza el programa shp2pgsql para cargar la información al PostgreSQL, o wkb_geometry si se utiliza ogr2ogr.

```
LAYER
  NAME pg_test
  TYPE POLYGON
  CONNECTIONTYPE POSTGIS
  CONNECTION "host=mapserver.com dbname=gmap user=leonardo"
  DATA "the_geom FROM province"
  CLASS
    ...
  END
END
```

WEB FEATURE SERVICE (WFS)

WFS es una especificación del Open Geospatial Consortium (OGC), mayor información sobre este formato se puede encontrar en <http://www.opengeospatial.org/>. Un WFS permite a un cliente recuperar información geoespacial codificada en Geography Markup Language (GML) de múltiples servicios web. El GML está construido sobre el lenguaje web XML estándar. El WFS difiere del popular Web Map Service (WMS) ya que el primero retorna un subconjunto de los datos en formato GML, mientras que el segundo solo genera una imagen de los datos.

Capacidades (Capabilities)

Al solicitar las Capacidades usando la petición `GetCapabilities` a un WFS se obtiene un documento XML que muestra todos los layers y proyecciones que este tiene disponibles, entre otras cosas.

Ejemplo de un URL `GetCapabilities`

Se debe ingresar este código **completo** en la barra de direcciones de un navegador web para que se produzca la respuesta deseada (en este ejemplo se muestra en varias líneas para facilitar su lectura).

```
http://www2.dmsolutions.ca/cgi-bin/mswfs_gmap
?VERSION=1.0.0
&SERVICE=wfs
&REQUEST=GetCapabilities
```

Ejemplo de respuesta XML de un `GetCapabilities`

```
...
  <FeatureTypeList>
    <Operations>
      <Query/>
    </Operations>
    <FeatureType>
      <Name>park</Name>
      <Title>Parks</Title>
      <SRS>EPSG:42304</SRS>
      <LatLongBoundingBox minx="-173.433" miny="41.4271" maxx="-
13.0481" maxy="83.7466" />
    </FeatureType>
    <FeatureType>
      <Name>road</Name>
      <Title>Roads</Title>
      <SRS>EPSG:42304</SRS>
      <LatLongBoundingBox minx="-148.059" miny="35.882" maxx="-
33.7745" maxy="72.5503" />
    </FeatureType>
    <FeatureType>
      <Name>popplace</Name>
      <Title>Cities</Title>
      <SRS>EPSG:42304</SRS>
      <LatLongBoundingBox minx="-172.301" miny="36.3541" maxx="-
12.9698" maxy="83.4832" />
    </FeatureType>
  </FeatureTypeList>
...
```

Acceso a la información

El acceso a WFS es uno de los atributos principales de MapServer. Actualmente MapServer soporta la versión WFS 1.0.0. el parámetro `ConnectionType WFS` es de uso obligatorio. Un layer WFS puede ser requerido / solicitado a través de un objeto `LAYER` en un mapfile, o se puede hacer la petición URL directamente desde el Web Browser por medio de la orden `GetFeature`. Se puede solicitar un layer específico con la orden `TypeName`. En un mapfile todas estas peticiones deben incluirse dentro de una estructura `METADATA`. Si se quiere mostrar en pantalla un layer con mucha información (que pueda traslaparse) con una escala grande, se puede limitar el numero de *features* que retorna el GML haciendo uso de la opción `MaxFeatures` (ej: `MaxFeatures = 100`).

Ejemplo de una solicitud WFS hecha directamente a través del Web Browser

La siguiente petición URL solicita el GML para el layer `road`. (ver el ejemplo anterior de petición `GetCapabilities` sobre los layers disponibles en este servidor de pruebas). El URL se debe ingresar en una sola línea, aquí se muestra en varias líneas para facilitar su lectura.

```
http://www2.dmsolutions.ca/cgi-bin/mswfs_gmap
?VERSION=1.0.0
&SERVICE=wfs
&REQUEST=getfeature&TYPENAME=road
```

Ejemplo de Mapfile

```
LAYER
  NAME "wfs_gmap_roads"
  STATUS DEFAULT
  TYPE LINE
  CONNECTIONTYPE WFS
  CONNECTION "http://www2.dmsolutions.ca/cgi-bin/mswfs_gmap?"
  METADATA
    "wfs_version" "1.0.0"
    "wfs_srs" "EPSG:42304"
    "wfs_typedname" "road"
    "wfs_request_method" "GET"
    "wfs_service" "WFS"
  END
  CLASS
    NAME "roads"
    COLOR 0 0 0
  END
END # layer
```

WEB MAP SERVER (WMS)

Un WMS (o Web Map Server) permite el uso de información - datos de un sinnúmero de servidores, y posibilita la creación en una red de servidores de mapas con la cual muchos usuarios puedan crear mapas personalizados.

Ejemplo de Mapfile

Se debe definir el parámetro `PROJECTION` en el Mapfile, a menos que se tenga la seguridad que todos los layers WMS de su aplicación tiene soporte para una misma proyección, que debe ser la misma que se define para tofo el mapa en el mapfile. La proyección puede ser definida usando el código `init=epsg:xxxx` o usando los parámetros `PROJ4`.

Debe especificarse el parámetro `IMAGEPATH` en el objeto `WEB` del Mapfile, y este debe apuntar a un directorio sobre el cual el servidor tenga propiedades de lectura / escritura. MapServer usa este directorio para almacenar archivos temporales descargados de los servidores remotos. Los archivos temporales son borrados automáticamente por MapServer.

```
LAYER
  NAME "prov_bound"
  TYPE RASTER
  STATUS ON
  CONNECTION "http://www2.dmsolutions.ca/cgi-bin/mwms_gmap?"
  CONNECTIONTYPE WMS
  METADATA
    "wms_srs"          "EPSG:42304 EPSG:42101 EPSG:4269 EPSG:4326"
    "wms_name"        "prov_bound"
    "wms_server_version" "1.1.0"
    "wms_formatlist"  "image/gif,image/png,image/jpeg,image/wbmp"
    "wms_format"      "image/gif"
  END
END
```

Los siguientes elementos son requeridos para que MapServer logre una conexión WMS:

CONNECTIONTYPE WMS Tipo de conexión.

CONNECTION Es el URL en línea del servidor remoto, solo se debe incluir la dirección base sin parámetros WMS. La versión del servidor, formato de imagen, nombre del layer, etc., se especifican por medio de `METADATA`.

wms_srs Es una lista de códigos EPSG de proyecciones soportadas por el servidor remoto. Normalmente estos se pueden obtener por medio de la propiedad `Capabilities` del servidor. Estos códigos deben estar en letras mayúsculas.

wms_name Es la lista de layers que son traídos del servidor remoto WMS.

wms_server_version Es la versión de protocolo WMS soportado por el servidor remoto WMS.

wms_format Es el formato de imagen usado en las peticiones `GetMap`.

BIBLIOGRAFÍA

Gran parte de este documento es una Traducción Libre del *MapServer Reference Guide for Vector Data Access*, de *Jeff McKenna y Tyler Mitchell - Febrero, 2005* que se encuentra disponible en la pagina web de MapServer <http://mapserver.gis.umn.edu/doc.html>,

La parte referida a Web Map Server (WMS) es una Traducción Libre del documento *MapServer WMS Client HOWTO* que se encuentra en la pagina web de MapServer en <http://mapserver.gis.umn.edu/doc/wms-server-howto.html>.

Para obtener mas información, favor remitirse al Trabajo de Grado *Implementación de Componentes Técnicos de la Infraestructura de Datos Espaciales Aplicados a Imágenes de Satélite Ortorectificadas: Prototipo de Clearinghouse*, de *Deiro González y Leonardo Montero – 2006, ANEXO D – EJEMPLO DE MAPFILE*.