

# earthFile

برای نوشتن یک earhfile از تگ های xml استفاده می شود که هر تگ باید باز و بسته شود  
</>

## تگ map

برای تعریف یک بستر از نقشه از که بالاترین سطح earhfile قرار میگیرد از تگ <map> استفاده می کنیم  
این تگ سه ویژگی دارد

1. name : string
2. type : geocentric -> render globe , projected -> render flat
3. version :2

در داخل این تگ میتوان تگ های زیر را نیز استفاده نمود

<options>  
<image>  
<elevation>  
<model>  
<mask>  
<libraries>

## تگ option

این تگ گزینه های و روش های نمایش یک نقشه و همچنین نحوه render شدن نقشه را کنترل می کند  
این تگ ۵ ویژگی نیز دارد.

1. lighting : true ,false-> میزان تاثیر روشنایی بر نقشه
2. elevation\_interpolation : الگوریتم مورد استفاده برای استفاده از منابع ارتفاع:
  - a. nearest
  - b. average
  - c. bilinear
  - d. biline
3. overlay\_texture\_size :
4. overlay\_blending : true , false -> ترکیب روکش با زمین
5. overlay\_resolution\_ratio : نسبت وضوح نزدیکی دوربین با زمین

داخل این تگ می شود تگ های را تعریف نمود

<profile>  
<proxy>  
<cache>  
<cache\_policy>  
<terrain>

## تگ profile

این تگ برای رندر کردن مختصات فضایی و کاشی نقشه مورد استفاده قرار میگیرد.

```
<profile srs = "+proj=utm +zone=17 +ellps=GRS80 +datum=NAD83 +units=m +no_defs"
```

نوع کد گذاری

```
vdatum = "egm96" داده های عمودی
```

```
xmin = "560725.500" گستره داده های نقشه
```

```
xmax = "573866.500"
```

```
ymin = "4385762.500"
```

```
ymax = "4400705.500"
```

```
num_tiles_wide_at_lod_0 = "1"
```

```
num_tiles_high_at_lod_0 = "1"> اندازه بیشترین مراتب کاشی
```

## تگ cache

برای تنظیمات cache سیستم

```
<cache driver = "filesystem" , level db
```

```
path = "c:/osgearth_cache" >
```

## تگ CachePolicy

مشخص میکند که یک آبجکت چگونه در یک آبجکت قرار بگیرد.

```
<cache_policy usage="no_cache" cache_only ,read_write
```

```
max_age = 12 ورودی های حافظه پنهان قدیمی تر از این مقدار (در ثانیه) را منقضی کنید
```

```
>
```

## تگ Proxy

تنظیمات پراکسی را مشخص می کند

```
<proxy host = "hostname"
```

```
port = "8080"
```

```
username = "jason"
```

```
password = "helloworld" >
```

## تگ image

این تگ یک لایه رستری از تصویر به ما میدهد

دارای چندین ویژگی نیز می باشد

```
<image name = "my image layer" -> string
```

```
driver = "gdal" -> پلاگین مورد استفاده برای یک لایه کاشی
```

```
nodata_image = "http://readymap.org/nodata.png" -> url
```

```
opacity = "1.0"
```

```
min_range = "0" -> حداقل محدود دید از دوربین اگر از این نزدیک تر شود کاشی دیده نمی شود
```

```
max_range = "100000000" -> حداکثر محدوده دید
```

```
attenuation_range = "0" فاصله بین حداکثر و کمترین محدوده دید
```

```
min_level = "0" کمترین سطح جزئیات
```

```
max_level = "23" بیشترین سطح جزئیات
```

```

min_resolution = "100.0"
max_resolution = "0.0"
max_data_level = "23"
enabled        = "true"
visible        = "true"
shared         = "false"
shared_sampler = "string"
shared_matrix  = "string"
coverage       = "false"
min_filter     = "LINEAR"
mag_filter     = "LINEAR"
blend          = "interpolate"
altitude       = "0"
texture_compression = "none" >

```

توضیحات و مقادیر هر کدام در داکيومنت اصلی نیز وجود دارد

در این تگ می شود دو تگ دیگر تعریف کرد

<cache\_policy>

<proxy>

## تگ Elevation

این تگ یک لایه شبکه ای ایجاد میکند و با استفاده از ارتفاع به صورت یک کاشی روی زمین قرار می دهد

```

<elevation name      = "text"
  driver             = "gdal"
  min_level          = "0"
  max_level          = "23"
  min_resolution     = "100.0"
  max_resolution     = "0.0"
  enabled            = "true"
  offset             = "false" -> نشان دهنده ارتفاع نسبت به زمین به صورت نسبی می باشد
  nodata_value       = "-32768"
  min_valid_value    = "-32768"
  max_valid_value    = "32768"
  nodata_policy      = "interpolate" -> msl . no data   مقادیری که داده ندارد را مشخص می کند

```

## تگ MODEL

این تگ برای استفاده از مدل های سه بعدی نیز استفاده می شود.

```

<model name  = "my model layer"
  driver     = "simple" >

```

با استفاده از MASK میتوان برشی از یک مدل را در زمین استفاده کرد

## تگ LIBRARY

کتابخانه‌ها در مسیر کتابخانه **osg** جستجو می‌شوند و نام کتابخانه باید از قرارداد نام کتابخانه **osg nodekit** پیروی کند با استفاده از ؛ می‌توان آنها را از هم جدا نمود

# Driver Reference

درایور یک ماژول پلاگین است که از برخی منابع خارجی در **osgEarth** پشتیبانی می‌کند.

## tile درایور

درایور **TileSource** درایوری است که داده‌های شطرنجی را برای موتور زمینی **osgEarth** فراهم می‌کند. می‌تواند کاشی‌های تصویر، کاشی‌های شبکه ارتفاعی یا هر دو را تولید کند.

## agglite

از کتابخانه **agglite** برای تبدیل داده‌ها به کاشی تصویر استفاده می‌شود و بیشتر از با تگ **feature** و داده‌های شیپ مورد استفاده قرار می‌گیرد

## arcgis

از سرورهای **arcgis** استفاده می‌کند. باید یک آدرس را در تگ **url** قرار داد.

```
<image driver="arcgis">
```

```
<url>http://services.arcgisonline.com/ArcGIS/rest/services/World_Imagery/MapServer</url>
</image>
```

## cesium ion

از سرویس‌های **cesium ion** استفاده می‌کند

```
<image name="cesiumion bluemarble" driver="cesiumion">
  <asset_id>3845</asset_id>
```

```
<token>eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOiI0NDViM2NkNi0xYTE2LTRlZTUtODBiNy05M2Q4ODg4M2NmMTQiLCJpZCI6ImJlU5LCJpYXQiOiE1MTgxOTc4MDh9.sld5jPORDf_IWavMEsugh6vHPnjR6j3qd1aBkQTswNM</token>
</image>
```

## color ramp

این پلاگین از یک فایل **ramp** رنگی از یک ارتفاع زیرین برای تولید تصاویر استفاده می شود

```
<image name="color ramp" driver="colorramp">
  <elevation name="readymap_elevation" driver="tms">
    <url>http://readymap.org/readymap/tiles/1.0.0/9/</url>
  </elevation>
  <ramp>..\data\colorramps\elevation.clr</ramp>
</image>
```

## debug

این افزونه طرح کلی یک کاشی را به همراه کلید کاشی ان نشان می دهد

```
<image driver="debug">
</image>
```

## gdal

اگر بخواهیم از فایل محلی خود استفاده کنیم از این درایور استفاده میکنیم

```
<image driver="gdal">
  <url>data/world.tif</url>
</image>
```

## mbtiles

اگر بخواهیم داده ها از پایگاه داده **sqlite** بخوانیم از این درایور استفاده می شود

```
<image name="haiti" driver="mbtiles">
  <filename>../data/haiti-terrain-grey.mbtiles</filename>
  <format>jpg</format>
</image>
```

## noise

با استفاده از یک مولد رویه ای نویز **perlin** رویه زمین را تولید می کند

```
<elevation driver="noise">
  <resolution>3185500</resolution>  <!-- 1/4 earth's diameter -->
  <scale>5000</scale>                <!-- vary heights by +/- 5000m over the resolution -->
  <octaves>12</octaves>              <!-- detail recursion level -->
</elevation>
```

## osg

برای بارگذاری تصاویر نیز استفاده می شود و سپس کاشی ها را بر اساس تصاویر برمیگرداند

```
<image driver="osg">
  <url>images/world.png</url>
  <profile>global-geodetic</profile>
</image>
```

## Quadkey

برای خواندن کاشی نقشه از سیستم نقشه bing استفاده می شود

```
<image name="imagery" driver="quadkey">
  <url>http://[1234].server.com/tiles/{key}.png</url>
</image>
```

## tilecache

برای ذخیره سازی کاشی ها استفاده می شود

```
<image driver="tilecache">
  <url>http://server/tiles/root</url>
  <layer>landuse</layer>
  <format>jpg</format>
</image>
```

## tilesevice

داده ها را از سرویس ناسا میخواند

```
<image driver="tileservice">
  <url>http://server/tileservice/tiles</url>
  <dataset>weather</dataset>
  <format>png</format>
</image>
```

## tms

این پلاگین با استفاده از سرویس های tms داده ها میگیرد

```
<image driver="tms">
  <url>http://readymap.org:8080/readymap/tiles/1.0.0/79/</url>
```

</image>

## vbp

این افزونه تلاش می کند تا کاشی های تصویر و ارتفاع را از مدل vbp استفاده بکند

```
<image driver="vpb">
  <url>http://www.openscenegraph.org/data/earth_bayarea/earth.ive</url>
  <profile>global-geodetic</profile>
  <primary_split_level>5</primary_split_level>
  <secondary_split_level>11</secondary_split_level>
  <directory_structure>nested</directory_structure>
</image>
```

## WCS

داده ها شطرنجی را به صورت محدود بر اساس مشخصات سرویس پوشش وب ocg می خواند و برای داده های ارتفاعی نیز مفید می باشد

```
<elevation driver="wcs">
  <url>http://server</url>
  <identifier>elevation</identifier>
  <format>image/GeoTIFF</format>
</elevation>
```

## wms

داده ها را از سرویس web map می خواند

```
<image name="Landsat" driver="wms">
  <url>http://onearth.jpl.nasa.gov/wms.cgi</url>
  <srs>EPSG:4326</srs>
  <tile_size>512</tile_size>
  <layers>global_mosaic</layers>
  <styles>visual</styles>
  <format>jpeg</format>
</image>
```

## xyz

داده ها را از وب با استاندارد x/y/old یز می خواند

```
<image name="mapquest_open_aerial" driver="xyz">
  <url>http://oatiles[1234].mqcdn.com/tiles/1.0.0/sat/{z}/{x}/{y}.jpg</url>
  <profile>spherical-mercator</profile>
</image>
```

## مدل درایور

درایور ModelSource درایوری است که یک گره OpenSceneGraph تولید می کند. osgEarth از ModelSources برای نمایش داده های ویژگی برداری و بارگذاری و نمایش مدل های سه بعدی خارجی استفاده می کند.

### feature\_geom

این افزونه داده های ویژگی برداری را با استفاده از استاندارد ها به هندسه OSG ارائه می کند.

```
<model driver="feature_geom">
  <features driver="ogr">
    <url>world.shp</url>
  </features>
  <styles>
    <style type="text/css">
      default {
        stroke:    #ffff00;
        stroke-width: 2;
      }
    </style>
  </styles>
  <fading duration="1.0"/>
</model>
```

### Simple Model

این افزونه به سادگی یک مدل سه بعدی خارجی را بارگیری می کند و به صورت اختیاری آن را در مختصات نقشه قرار می دهد.

```
<model name ="model" driver="simple">
  <url>../data/red_flag.osg.100,100,100.scale</url>
  <location>-74.018 40.717 10</location>
</model>
```

## درایور ویژگی ها

درایور ویژگی پلاگینی است که داده های بردار نسبت داده شده را می خواند

### ogr

این افزونه داده های برداری را از هر یک از فرمت های پشتیبانی شده توسط OGR Simple Feature Library (که بسیار زیاد است) می خواند. رایج ترین آنها شامل GML، ESRI Shapefiles و PostGIS است.



```
<model driver="feature_geom">
  <features driver="ogr">
    <url>data/world_boundaries.shp</url>
  </features>
```

## tfs

این پلاگین داده های برداری را از مخزن خدمات Tiled Feature می خواند. TFS یک طرح کاشی کاری شده شبیه به TMS (سرویس نقشه کاشی) است اما برای داده های ویژگی برش داده شده است.

```
<model driver="feature_geom">
  <features driver="tfs">
    <url>http://readymap.org/features/1/tfs/</url>
    <format>json</format>
  </features>
```

## wfs

پلاگینی برای خواندن داده های وکتوری از مخازن wfs می باشد

```
<model driver="feature_geom">
  <features name="states" driver="wfs">
    <url> http://demo.opengeo.org/geoserver/wfs</url>
    <typename>states</typename>
    <outputformat>json</outputformat>
  </features>
```

## mapnikvectortiles

این پلاگین داده های وکتوری را از mbtile میگیرد

این درایور در حال حاضر از فایل های mbtiles چند سطحی پشتیبانی نمی کند. این فقط حداکثر سطح را در پایگاه داده بارگذاری می کند. زمانی که osgEarth از منابع داده ویژگی های غیرافزودنی پشتیبانی بهتری داشته باشد، این در آینده تغییر خواهد کرد.

```
<model driver="feature_geom">
  <features name="osm" driver="mapnikvectortiles">
    <url>../data/osm.mbtiles</url>
  </features>
```

## درایور terrain

A Terrain Engine Driver افزونه ای است که زمین osgEarth را رندر می کند. در بیشتر موارد، باید از پیش فرض استفاده کنید - اما افزونه های موتور زمین قدیمی برای پشتیبانی موقت از کاربردهایی که هنوز نیاز به انتقال به جدیدترین نسخه osgEarth دارند

### mp

موتور پیش فرض زمین برای osgEarth تعداد نامحدودی از لایه های تصویر را با استفاده از تکنیک ترکیب چند پاسی در سطح کاشی ارائه می کند.

```
<map>
  <options>
    <terrain driver      = "mp"
      skirt_ratio       = "0.05"
      color              = "#ffffff"
      normalize_edges   = "false"
      incremental_update = "false"
      quick_release_gl_objects = "true"
      min_tile_range_factor = "7.0"
      cluster_culling    = "true" />
```

## درایور effect

پلاگین هایی که جلوه های ویژه را پیاده سازی می کنند.

### Gl sky

یک مدل آسمان است که میتوان زمان را در آن مشخص نمود.

```
<map>
  <options>
    <sky driver = "gl"
      hours = "0.0"
      ambient = "0.05" />
```

### simple sky

یک مدل آسمان است پراکنندگی و روشنایی اتمسفر را اجرا می کند.

```
<map>
  <options>
    <sky driver      = "simple"
      hours          = "0.0"
      ambient        = "0.05"
```

```
atmospheric_lighting = "true"
exposure               = "3.0" />
```

## silverlining

```
<map>
  <options>
    <sky driver = "silverlining"
      hours      = "0.0"
      ambient    = "0.05"
      user       = "myname"
      license_code = "mycode"
      clouds     = "false"
      clouds_max_altitude = "0.0" />
```

## درایور cache

افزونه ای است که کاشی های زمین و قابلیت ذخیره اطلاعات را در دیسک محلی فراهم می کند.

## file system cache

این افزونه کاشی های زمین، بردارهای ویژگی و سایر داده ها را در سلسله مراتبی از پوشه ها در سیستم فایل محلی ذخیره می کند. هر عنصر داده ذخیره شده در یک فایل جداگانه است و ممکن است شامل یک فایل فراداده مرتبط باشد.

```
<map>
  <options>
    <cache driver="filesystem">
      <path>c:/osgearth_cache</path>
    </cache>
```

## leveldb

این افزونه کاشی های زمین، بردارهای ویژگی و سایر داده ها را با استفاده از کتابخانه ذخیره کلید/مقدار تعبیه شده Google leveldb در سیستم فایل محلی ذخیره می کند. حافظه پنهان leveldb هر دسته از داده ها را در سطل مخصوص خود ذخیره می کند. همه bin ها در یک فهرست، در یک پایگاه داده ذخیره می شوند.

```
<map>
  <options>
    <cache driver    = "leveldb"
      path          = "c:/osgearth_cache"
      max_size_mb   = "500" />
  </cache>
```

پایان