

# 2-tägige MapServer In-House Schulung für das LGLN SLA

Björn Schilberg  
Intevation GmbH

<http://www.intevation.de/geospatial>  
[bjoern.schilberg@intevation.de](mailto:bjoern.schilberg@intevation.de)

10.12.2012 & 11.12.2012  
Hannover

## Überblick

- Merkmale
- Community
- Historie
- Anatomie einer MapServer Anwendung
- Themen der Schulung

## Einrichtung (HandsOn)

- Installation des MapServers unter SLES.
- Integration des MapServers in den Apache HTTP Server.
- Integration des MapServers als CGI-Modul.
- Integration des MapServers als FastCGI-Modul.

## Konfiguration

- Konfiguration über das MAP-File.
- Erläuterung der MapServer-Syntax.
- **Übung:** Erzeugen Sie Ihr erstes Kartenbild.
- Aufsetzen eines WMS-Dienstes.
- Einführung in die Möglichkeiten der Fehlersuche.

## Tipps und Tricks

- **Übung:** Frida-WMS
- Symboliken und Klassifizierungen erstellen.
- MapServer als WMS Client (Kasade).
- Proxy Konfiguration.
- MapServer als WFS Server.
- Performance-Tipps bei großen Datenmengen (Mit Beispieldaten von LGLN SLA?).

## Einfache WebClient-Applikationen mit MapServer erstellen

- Vorstellung der WebMapping-Clients OpenLayers und Leaflet.
  - MapServer Integration am Beispiel von OpenLayers.
  - MapServer Integration am Beispiel von Leaflet.

# Merkmale des MapServers

- Software zur Erstellung interaktiver Karten und mehr.
- Unterstützung zahlreicher OGC Standards (WMS, WFS, WCS, WMC, WMS Time, SOS, SLD).
- Als CGI- oder FastCGI-Programm lauffähig.
- Kann Templates zur Anzeige nutzen.
- Verfügt über eine eigene Programmierschnittstelle namens MapScript.
  - Unterstützung für Skriptsprachen und Entwicklungsumgebungen (PHP, Python, Perl, Ruby, java, .NET).
- Konfiguration über map-Dateien.
- Unterstützung zahlreicher Vektor und Rasterformate.



- Über Jahre gewachsenes und sehr erfolgreiches Projekt.
- MapServer wird weltweit genutzt.
- Große deutschsprachiger Anwendergemeinschaft.
- Seit Jahren erfolgreich im produktiven Einsatz.
- Wachsende deutschsprachige Dokumentation.
  - <http://mapserver.org/de/index.html>
- Deutschsprachiger Support über die MailingListe.
  - [talkliste@fossgis.de](mailto:talkliste@fossgis.de)
- Seit Jahren mit Vorträge und Workshops auf der FOSSGIS Konferenz vertreten.



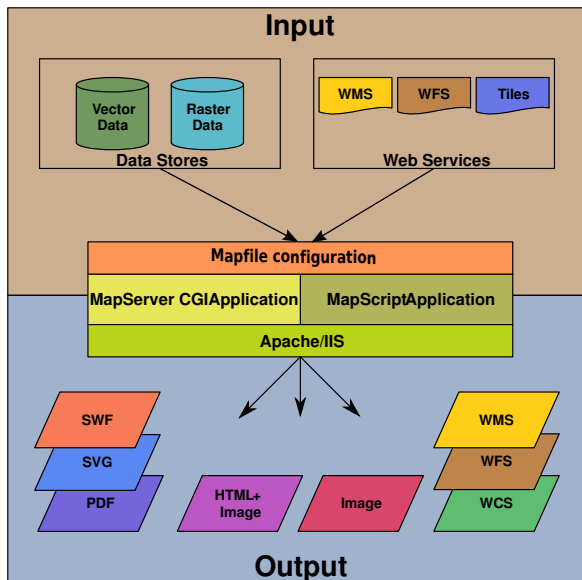
- 1916 registrierte Mailadressen bei MapServer-users
- 403 bei der MapServer-dev Liste
- 493 bei der ms4w-users Liste
- 12 Mitglieder im Project Steering Committee (PSC)
- 25 aktive Committer im MapServer-Projekt
- 3000-5000 mal pro Monat wird das ms4wPaket heruntergeladen
- IRC #mapserver: Tummeln sich 30 Aktive
- 227,976 lines of code
- <http://www.ohloh.net/p/mapserver>

## History



Quelle: Präsentation FOSS4G 2010 Barcelona: <http://2010.foss4g.org/presentations/3370.pdf>

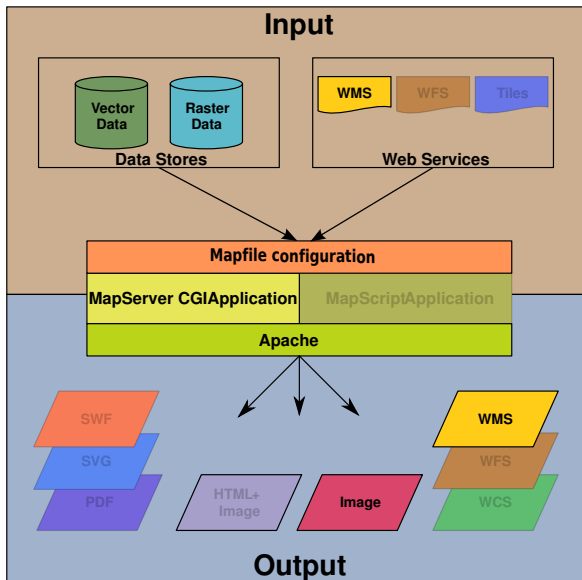
# Anatomie einer MapServer Anwendung



Quelle: <http://mapserver.org/de/introduction.html#anatomie-einer-mapserver-anwendung>



# Themen der Schulung



- <http://adminton.wald.intevation.org/>
- Adminton dient dem Führen von Logbüchern.
- Dokumentation während der gesamten Installation und Konfiguration.
- `/etc/logbuch.txt`
- Wird von unten nach oben gelesen.
- Aufruf mit `log`

## Installation von Adminton<sup>a</sup>

<sup>a</sup>Auszug aus /etc/logbuch.txt

```
06.12.2012      root
* curl -O http://hg.intevation.org/adminton/raw-file/tip/logbuch-tools/logbuch-
  installer
  bash logbuch-installer
```

## Einrichtung und Test der benötigten Paket-Repositories<sup>a</sup>

<sup>a</sup>Auszug aus /etc/logbuch.txt

```
07.12.2012      root
* zypper lr -d
  zypper se -s mapserver

07.12.2012      root
* zypper ar http://download.opensuse.org/repositories/home:/intevation:/sla/
  SLE_11_SP2/home:intevation:sla.repo
  rpm --import http://download.opensuse.org/repositories/home:/intevation:/sla/
  SLE_11_SP2/repodata/repomd.xml.key
  zypper ref

07.12.2012      root
* zypper ar http://download.opensuse.org/repositories/Application:/Geo/
  SLE_11_SP2/Application:Geo.repo
  rpm --import http://download.opensuse.org/repositories/Application:/Geo/
  SLE_11_SP2/repodata/repomd.xml.key
  zypper ref

07.12.2012      root
* zypper ar cd:////?devices=/dev/sr0 "SUSE Linux Enterprise Software Development
  Kit 11 SP2"
  zypper ar cd:////?devices=/dev/sr1 "SUSE-Linux-Enterprise-Server-11-SP2
  11.2.2-1.234"
```

## Installation des Mapserver<sup>a</sup>

<sup>a</sup>45 new packages to install. Overall download size: 27.3 MiB. After the operation, additional 94.9 MiB will be used.

```
07.12.2012      root
* zypper in mapserver
* zypper in proj
```

## Erste Funktionstests

```
07.12.2012      root
* /srv/www/cgi-bin/mapserv -v
  MapServer version 6.2.0 OUTPUT=GIF OUTPUT=PNG OUTPUT=JPEG OUTPUT=KML SUPPORTS=
  PROJ SUPPORTS=GD SUPPORTS=AGG SUPPORTS=FREETYPE SUPPORTS=CAIRO SUPPORTS=
  ICONV SUPPORTS=FRIBIDI SUPPORTS=WMS_SERVER SUPPORTS=WMS_CLIENT SUPPORTS=
  WFS_SERVER SUPPORTS=WFS_CLIENT SUPPORTS=WCS_SERVER SUPPORTS=SOS_SERVER
  SUPPORTS=FASTCGI SUPPORTS=THREADS SUPPORTS=GEOS INPUT=JPEG INPUT=POSTGIS
  INPUT=OGR INPUT=GDAL INPUT=SHAPEFILE
```

## Apache HTTP Server starten und erster Funktionstest

```
07.12.2012      root
* /etc/init.d/apache2 start
  curl http://localhost/cgi-bin/mapserv
  No query information to decode. QUERY_STRING is set , but empty.
```

# Integration des MapServers als CGI-Modul

Verschiedene Arten der Integration möglich:

## Bekannte Arten

- Wrapper script
  - `http://mapserver.org/de/ogc/wms_server.html?highlight=virtualhost#wrapper-script-unix`
- Apache SetEnvIf
  - `http://mapserver.org/de/ogc/wms_server.html?highlight=virtualhost#apache-setenvif`
- Apache environment variables - MS\_MAPFILE
  - `http://mapserver.org/de/ogc/wms_server.html?highlight=virtualhost#apache-environment-variables-ms-mapfile`

## Weniger bekannte Arten

- Python Wrapper script
- ASP script (IIS - Microsoft Windows)



# Integration des MapServers als FastCGI-Modul

## Auszug aus der Apache VirtualHost-Konfigurationsdatei<sup>a</sup>

<sup>a</sup>/etc/apache2/vhosts.d/

```
# ScriptAlias: This controls which directories contain server scripts.
# ScriptAliases are essentially the same as Aliases, except that
# documents in the realname directory are treated as applications and
# run by the server when requested rather than as documents sent to the client.
# The same rules about trailing "/" apply to ScriptAlias directives as to
# Alias.
#
ScriptAlias /fcgi-bin/ "/srv/www/fcgi-bin/"

# "/srv/www/fcgi-bin" should be changed to whatever your ScriptAliased
# CGI directory exists, if you have one, and where ScriptAlias points to.
#
<Directory "/srv/www/fcgi-bin">
SetHandler fcgid-script
SetEnvIf Request_URI "/fcgi-bin/sample.fcgi" MS_MAPFILE=/PATH-TO-MAPFILE/sample.
    map
AllowOverride None
Options +ExecCGI -MultiViews FollowSymLinks
Order allow,deny
Allow from all
</Directory>
```

Weiterführende Dokumentation: <http://mapserver.org/optimization/fastcgi.html?highlight=fastcgi>

## Syntax-Highlight-Unterstützung für den Editor vim einrichten<sup>a</sup>

<sup>a</sup>Auszug aus /etc/logbuch.txt

```
06.12.2012      root
* curl http://mapserver.org/de/_static/map.vim -o /usr/share/vim/current/syntax/
  map.vim
```

Dokumentation:

<http://mapserver.org/de/development/editing/vim.html>

## Einfache Mapdatei mit einer Ebene

```
MAP
  NAME "sample"
  STATUS ON
  SIZE 800 400
  EXTENT -180 -90 180 90
  SHAPEPATH "data/"

  WEB
    IMAGEPATH "tmp/"
  END

  LAYER
    NAME 'global-raster'
    TYPE RASTER
    STATUS DEFAULT
    DATA bluemarble.tif
  END
END
```

- Konfigurationsdatei mit strukturiertem Text (ASCII-Textdatei).
- Datenquellen definiert.
- Gebiet der Karte definiert.
- Aussehen der Karte definiert.

## Einfache Mapdatei mit einer Ebene

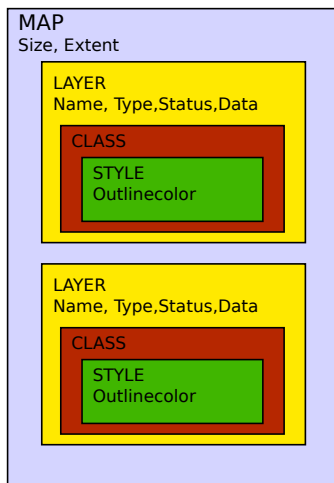
```
MAP
  NAME "sample"
  STATUS ON
  SIZE 800 400
  EXTENT -180 -90 180 90
  SHAPEPATH "data/"

  WEB
    IMAGEPATH "tmp/"
  END

  LAYER
    NAME 'global-raster'
    TYPE RASTER
    STATUS DEFAULT
    DATA bluemarble.tif
  END
END
```

- Besteht aus Objekten.
- Objekt-Bereiche/Blöcke werden mit END abgeschlossen.
- Jedes Objekt hat eine Vielzahl von Parametern.
- Alle Objekte mit deren Parameter sind in der [Mapfile Referenz](#) dokumentiert.

# Aufbau des MAP-Files<sup>1</sup>



Quelle: Web Mapping Illustrated Using  
Open Source GIS Toolkits By Tyler  
Mitchell, Seite 155

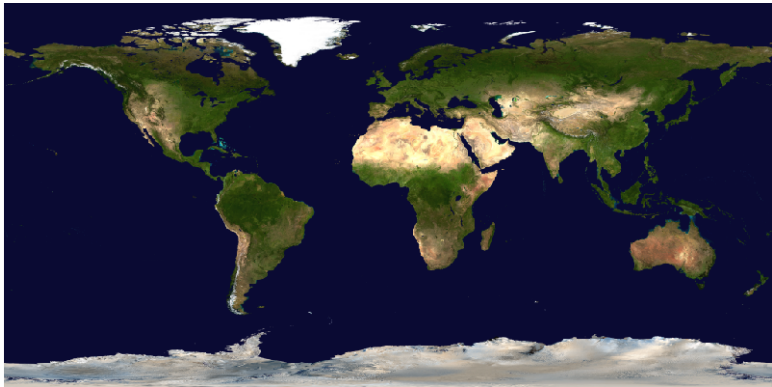
- Das MAP-Objekt enthält projektweite Einstellungen.
- Das LAYER-Objekt definiert eine Ebene (Layer).
- Das CLASS-Objekt definiert eine Klasse über die bestimmte Objekte dargestellt werden sollen.
- Das STYLE-Objekt definiert wie Objekte einer bestimmten Klasse dargestellt werden sollen.

# Regeln und Tipps für den Aufbau der Mapdatei

- Kommentare beginnen mit der Raute #.
- MapServer arbeitet das Mapfile von oben nach unten ab, Layer am Ende der Datei werden zum Schluß gezeichnet. D.h. diese werden auf alle vorherigen Layer gezeichnet.
- Es sollten immer relative Pfade benutzt werden (Relativ zur Mapdatei).
- Pfade sollten immer maskiert werden (einzelne oder doppelte Hochkommata werden akzeptiert).
- Einrückung nicht vorgeschrieben, aber empfohlen (leichter lesbar).
- Objekt- und Parameter-Namen sind nicht case-sensitive. Empfehlung Angabe in Großbuchstaben.
- Attribute von Felder von Datenquellen und Pfadangaben case-sensitive angeben! Auf Groß- und Kleinschreibung achten!
- Mapdateien müssen mit der Endung `.map` versehen werden.

- Feststehende Schlüsselwörter ohne Anführungszeichen, z.B.
  - STATUS ON
  - TYPE RASTER
- Variable Zeichenketten mit Anführungszeichen, z.B.
  - NAME 'global-raster'
- Zahlenwerte ohne Anführungszeichen, z.B.
  - IMAGECOLOR 255 255 255

# Übung: Erzeugen Sie Ihr erstes Kartenbild





# Übung: Erzeugen Sie Ihr erstes Kartenbild

## Mapdatei sample.map mit einer Ebene

```
MAP
  NAME "sample"
  STATUS ON
  SIZE 800 400
  EXTENT -180 -90 180 90
  SHAPEPATH "data/"

  WEB
    IMAGEPATH "tmp/"
  END

  LAYER
    NAME 'global-raster'
    TYPE RASTER
    STATUS DEFAULT
    DATA bluemarble.tif
  END
END
```

Datensatz: <ftp://ftp.intevation.de/geospatial/geospatial/lgl-n-sla/>

## Verzeichnis Aufbau

```
├── sample
│   ├── data
│   │   ├── bluemarble.tif
│   │   └── sample.map
│   └── tmp
```

# Übung: Erzeugen Sie Ihr erstes Kartenbild

Erzeugung einer Bilddatei mittels `shp2img`

```
shp2img -m sample.map -o erste_karte.png
```

Aufruf eines dynamischen Kartenbildes mittels `MapServer`  
`OpenLayers Viewer`

```
http://<servername>/cgi-bin/mapserv?MAP=<path-to-mapfile>/sample/sample.map&mode=browse&template=openlayers
```

## GetCapabilities-Abruf

```
http://<servername>/cgi-bin/mapserv?MAP=/home/  
intevation/mapservers/sample/sample.map&REQUEST=  
GetCapabilities&SERVICE=WMS&VERSION=1.1.1
```

# Aufsetzen eines WMS Dienstes

```
MAP
  NAME "sample"
  STATUS ON
  SIZE 800 400
  EXTENT -180 -90 180 90
  SHAPEPATH "data/"

  WEB
    METADATA
      "wms_title" "Sample"
      "wms_onlineresource" "http://<servername>/cgi-bin/mapserv?map=/home/
        intevation/mapserver/sample/wms.map"
      "wms_srs" "EPSG:4326 EPSG:3785 EPSG:3857 EPSG:900913"
      "wms_enable_request" "*"
    END
  END

  PROJECTION
    "init=epsg:4326"
  END

  LAYER
    NAME 'global-raster'
    METADATA
      "wms_title" "global-raster"
    END
    TYPE RASTER
    STATUS ON
    DATA bluemarble.tif
  END
END
```

## GetCapabilities-Abruf (Version 1.1.1)

```
http://<servername>/cgi-bin/mapserv?MAP=/home/  
intevation/mapserver/sample/wms.map&REQUEST=  
GetCapabilities&SERVICE=WMS&VERSION=1.1.1
```

## GetMap-Abruf (Version 1.1.1)

```
http://<servername>/cgi-bin/mapserv?map=/home/  
intevation/mapservers/sample/wms.map&SERVICE=WMS  
&VERSION=1.1.1&REQUEST=GetMap&BBOX  
=-180,-90,180,90&SRS=EPSG:4326&WIDTH=600&HEIGHT  
=400&LAYERS=global-raster&STYLES=&FORMAT=image/  
png&TRANSPARENT=TRUE
```

## GetMap-Abruf (Version 1.3.0)

```
http://<servername>/cgi-bin/mapserv?map=/home/  
intevation/mapservers/sample/wms.map&SERVICE=WMS  
&VERSION=1.3.0&REQUEST=GetMap&BBOX  
=-90,-180,90,180&CRS=EPSG:4326&WIDTH=600&HEIGHT  
=400&LAYERS=global-raster&STYLES=&FORMAT=image/  
png&TRANSPARENT=TRUE
```

[http://mapserver.org/de/ogc/wms\\_server.html](http://mapserver.org/de/ogc/wms_server.html)

- MapServer-Logging
  - Ausgaben umlenken.
    - `CONFIG "MS_ERRORFILE" "logs/ms_error.log"`
    - <http://mapserver.org/de/development/rfc/ms-rfc-28.html#setting-ms-errorfile>
- MapServer-Debugging
  - Über den Parameter `DEBUG` kann das Debugging des MapServers aktiviert werden.
  - Kann auf Ebene des MAP- als auch auf Ebene des LAYER-Objektes gesetzt werden.
  - <http://www.mapserver.org/de/optimization/debugging.html#step-2-set-the-debug-level>
- Webserver-Logs überprüfen
  - `/var/log/apache2/*`
  - <http://www.mapserver.org/de/optimization/debugging.html#step-6-check-your-web-server-logs>



## Tipps und Tricks

- **Übung:** Frida-WMS
- Symboliken und Klassifizierungen erstellen.
- MapServer als WMS Client (Kasade).
- Proxy Konfiguration.
- MapServer als WFS Server.
- Performance-Tipps bei großen Datenmengen (Mit Beispieldaten von LGLN SLA?).

## Einfache WebClient-Applikationen mit MapServer erstellen

- Vorstellung der WebMapping-Clients OpenLayers und Leaflet.
  - MapServer Integration am Beispiel von OpenLayers.
  - MapServer Integration am Beispiel von Leaflet.

# Übung: Frida-WMS



# Übung: Frida-WMS

- Laden Sie sich die Daten von <http://frida.intevation.de/index.html> herunter.
  - `frida-1.0.1-shp-joined.tar.gz` (650 KByte, Attribut-Klassen direkt angebunden)
- Machen Sie sich mit den Daten vertraut.
- Erstellen Sie einen Frida WMS.
- Erstellen Sie zu jedem Datensatz einen eignen Layer (gewaesserflaechen, gewaesserlinien , gruenflaechen, strassen, poi).
- Unterstützen Sie die Projektionen: EPSG:31467 EPSG:4326 EPSG:3785 EPSG:3857 EPSG:900913.
- Machen Sie sich mit den Eigenschaften CLASS, STYLE und COLOR vertraut.
- Wählen Sie geeignete Farben für die Objekte.
- Testen Sie das Mapfile und den Dienst ausführlich.

## Parkplatz-Symbol

```
SYMBOL  
  NAME "parkplatz"  
  TYPE pixmap  
  IMAGE images/  
    parkplatz.png  
END
```

## Quadrat

```
SYMBOL  
  NAME 'andere'  
  TYPE VECTOR  
  FILLED true  
  POINTS  
    0 1  
    0 0  
    1 0  
    1 1  
    0 1  
  END  
END
```

# Weitere Infos zur Symbolgenerierung

- <http://www.mapserver.org/de/mapfile/symbology/construction.html>
- <http://www.mapserver.org/de/mapfile/symbology/examples.html>
- <http://www.mapserver.org/de/mapfile/symbol.html>

## Erstellung einer Klassifizierung für Parkplätze

```
CLASS
NAME "Parkplatz"
EXPRESSION ('[poiTypeID]'='3')
SYMBOL "parkplatz"
  STYLE
    COLOR 0 0 0
    SIZE 25
    MAXSIZE 20
    MINSIZE 10
  END
END
```

Dokumentation: <http://www.mapserver.org/de/mapfile/expressions.html#expressions>

- Erstellen Sie eine Klasse für Parkhäuser (' [poiTypID] '=' 10').
- Erstellen Sie eine Klasse für Kliniken/Hospitale (' [poiTypID] '=' 2').
- Erstellen Sie eine Klasse für die poiTypIDs 4,5 und 9 (Öffentliche Gebäude).

## Erweiterung des Mapfiles um einen RemoteWMS-Layer

```
LAYER
  NAME "RemoteWMS"
  TYPE RASTER
  STATUS ON
  CONNECTION "http://demo.intevation.de/cgi-bin/frida-wms"
  CONNECTIONTYPE WMS
  EXTENT 3427000.290000 5787594.240000 3444004.000000
        5800876.470000
  PROJECTION
    "init=epsg:31467"
  END
  METADATA
    "wms_name"           "gruenflaechen"
    "wms_server_version" "1.1.1"
    "wms_format"        "image/png"
  END
END
```



# Übung: MapServer als WMS Client (Kaskade)

- Ersetzen Sie den Layer `gruenflaechen` aus dem vorherigen Beispiel durch den `RemoteWMS` aus der vorherigen Folie.

## Proxy-Konfiguration im WEB-Teil des MAP- oder LAYER-Blocks

```
WEB
  METADATA
    ...
    "wms_proxy_host" "127.0.0.1"
    "wms_proxy_port" "8000"
    ...
  END
END
```

Ein WFS-Dienst kann zwei Ausprägungen besitzen:

- Basis WFS
  - lesender Zugriff auf die Daten des WFS, mittels der Aufrufe:
    - `GetCapabilities`
    - `DescribeFeatureType` (Beschreibung des Features holen)
    - `GetFeature` (Das benannte Feature holen)
- Transaktionaler WFS (WFS-T)
  - lesender **und** schreibender Zugriff auf die Daten des WFS, mittels folgender zusätzlicher Aufrufe:
    - `Transaction` (Update, Insert, Delete)
    - `LockFeature`, `GetFeatureWithLock`

Zusammenfassung: GeoObjekte können im Falle des WFS gelesen werden. Im Falle von WFS-T können Geo-Objekte bearbeitet, erzeugt oder gelöscht werden.

## WFS-Konfiguration im MAP-Block

```
WEB
  METADATA
    "wfs_title"           "Osnabrueck"
    "wfs_onlineresource" "http://localhost:8080/cgi-bin/frida.cgi"
    "wfs_enable_request" "*"
    "wfs_srs"             "EPSG:31467 EPSG:4326 EPSG:3785 EPSG:3857 EPSG:900913"
    "wms_title"           "Osnabrueck"
    "wms_onlineresource" "http://localhost:8080/cgi-bin/frida.cgi"
    "wms_srs"             "EPSG:31467 EPSG:4326 EPSG:3785 EPSG:3857 EPSG:900913"
    "wms_enable_request" "*"
  END
END
```

## WFS-Konfiguration im LAYER-Block

```
DUMP TRUE
METADATA
  "wfs_title"           "gewaesserflaechen"
  "wms_title"           "gewaesserflaechen"
  "gml_include_items"  "all" ## Optional (serves all attributes for layer)
  "gml_featureid"     "gpShapelD" ## REQUIRED
END
```

# Übung: WFS-Erweiterung und Tests

- Speichern Sie das Mapfile aus der letzten Übung unter einem neuen Namen.
- Erweitern Sie das Mapfile um die WFS-Unterstützung.
- Testen Sie den Frida WFS mit folgenden Aufrufen im Google-Chrome oder Firefox.

## GetCapabilities:

Request=GetCapabilities&Version=1.0.0&Service=WFS

## GetFeature (alle Feature):

Request=GetFeature&Version=1.0.0&SERVICE=WFS&TYPENAME=poi

## GetFeature (max. 2 Features):

Request=GetFeature&Version=1.0.0&SERVICE=WFS&TYPENAME=poi&maxfeatures=2

## DescribeFeatureType:

Request=DescribeFeatureType&Version=1.0.0&SERVICE=WFS&TYPENAME=poi

## Räumlicher Filter BBOX (WFS 1.0.0) / Datenausgabe räumlich Beschränken:

Request=GetFeature&Version=1.0.0&SERVICE=WFS&TYPENAME=poi

&BBOX=3434752.611565,5794083.223668,3435206.275945,5794387.808535

## GetFeature (Wieviel Features sind verfügbar (Erst ab WFS 1.1.0)):

Request=GetFeature&Version=1.1.0&SERVICE=WFS&TYPENAME=poi&resulttype=hits

## GetFeature (Umprojektion (Erst ab WFS 1.1.0)):

Request=GetFeature&Version=1.1.0&SERVICE=WFS&TYPENAME=poi&maxfeatures=1&srsName=EPSG:4326

- Beispiele
  - `http://geoservices.knmi.nl/adaguc_portal/index.html`
    - Layer RADNL 25PCPRR L3 - Precipitation auswählen.
  - `http://mapstory.org/maps/153/embed`
- MapServer Dokumentation:  
`http://mapstory.org/maps/153/embed`

- Datenaufbereitung und Optimierung für Shapefiles
  - Indizierung: große Shapefiles sollten einen Index haben (`shptree <shapefile>`).
- Datenaufbereitung und Optimierung für Rasterdaten
  - Kachelung
  - Bildkataloge (Tileindex)
  - Rasterübersichten (Overviews/Pyramiden)



- Indizierung von Shapefiles
  - Erzeugung eines Quadtree-Indexes (.qix).
  - Einbindung der Shapes im Mapfile ohne .shp- Endung
  - MapServer nutzt diesen Index, um Objekte schnell zu finden.
  - shptree <shape>  
Creating index of new LBSformat
  - <http://mapserver.org/de/utilities/shptree.html>

# Übung: Optimierung von Vektordaten

- Erzeugen Sie für den zuvor in den Übungen erstellen Frida WMS Quadtree-Indices.

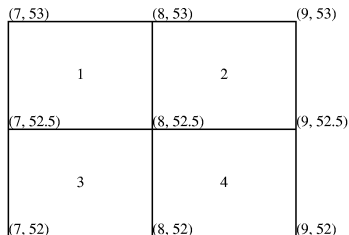
- Erstellung von Rasterübersichten (Overviews/Pyramiden) mittels gdal.
  - Mapstabsabhängige Übersichtskarten für ein Rasterbild schreiben.
  - Anwendung auch über eine Schleife, wenn viele Datensätze vorhanden sind.
  - Vorsicht: Die Originaldaten werden verändert!
  - `gdaladdo -r average <file> 2 4 8 16`
  - <http://www.gdal.org/gdaladdo.html>

# Optimierung von Rasterdaten mittels Kachelung

- Erstellung von Kachelung mittels gdal.
  - Rasterdateninformationen mit `gdalinfo`.
  - Zerlegung von großen Bildern in kleine Kacheln mit `gdal_translate`.
  - Anwendung von Schleifen bei großen Datenmengen sinnvoll.
  - `gdal_translate -projwin 7 53 8 53.5 infile.tif outfile.tif`
  - [http://www.gdal.org/gdal\\_translate.html](http://www.gdal.org/gdal_translate.html)

# Übung: Optimierung von Rasterdaten mittels Kachelung

Erzeugen Sie aus `satelit-os.tif` Kacheln in der Reihenfolge 1,2,3,4:



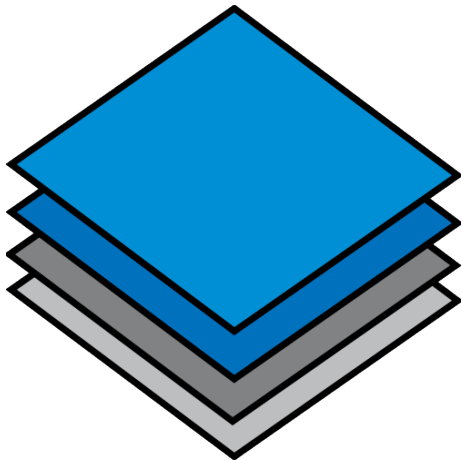
## Verwendung von `gdal_translate`

```
gdal_translate -projwin 7 53 8 52.5 satelit-os.tif sat1.tif  
gdal_translate -projwin 8 53 9 52.5 satelit-os.tif sat2.tif  
gdal_translate -projwin 7 52.5 8 52 satelit-os.tif sat3.tif  
gdal_translate -projwin 8 52.5 9 52 satelit-os.tif sat4.tif
```

- Erstellung von Bildkataloge mittels gdal-Werkzeugen.
  - Erzeugung eines Shapefile mit den Umrissen (BBOX) und Verweise auf die Datensätze im Dateisystem.
  - Ein neues Shape wird angelegt, sofern noch keines da ist. Ansonsten wird angehängt.
  - Projektion wird übernommen.
  - `gdaltindex <shapefile> <rasterdatei>`
  - <http://www.gdal.org/gdaltindex.html>

# Es geht noch vieles mehr ...

- Abfrage von Sachdaten (GetFeatureInfo) und deren Präsentation in individuellen html-Dateien.
- Bereitstellung von Legenden (GetLegendGraphic).
- Datenbanken oder andere Datenquellen.
- Bereitstellung verschiedener STYLES für denselben Layer.
- Maßstabsabhängige Darstellungen.
- Verwendung von SLDs.
- INSPIRE View Service im MapServer.
- Konvertierung von ESRI MXD Dateien nach Mapfiles.  
(<http://mxd2map.org>)
- ...



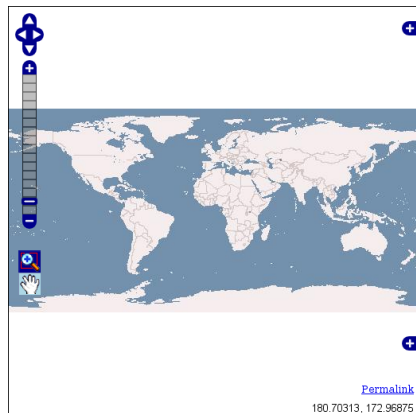


# Was ist OpenLayers?

- Freie API um dynamische Karten **in beliebigen Webseiten** zu integrieren.
- **Ausschließlich klientseitiges JavaScript**
- Unterstützung von freien und offenen Layern
- Außerdem Unterstützung von restriktiven Layern
- Benutzt Kachelung (Tiling) zur Darstellung
- Freie Software (BSD Lizenz)

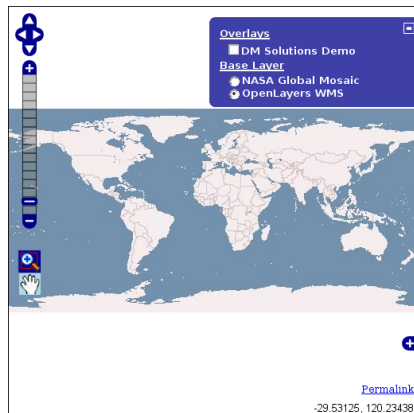
- Begonnen nach der Where 2.0 im Jahr 2005
- Motiviert durch MetaCarta's Geschäftsideen
- Erste Version veröffentlicht vor der Where 2.0 im Jahr 2006

Und so kann's aussehen . . .



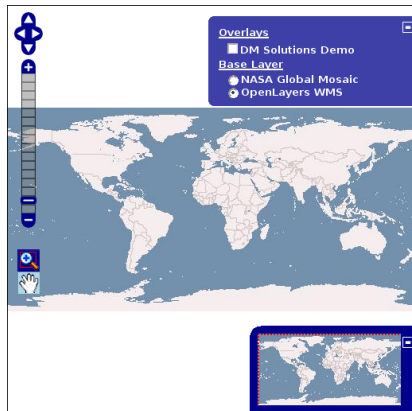
Openlayers in Action?!

# Und so kann's aussehen . . .



Openlayers in Action?!

# Und so kann's aussehen . . .



Openlayers in Action?!

## freie Layer

- OGC WMS
- OGC WFS
- GeoRSS
- CSV
- ka-Map
- WorldWind (\*)
- Canvas

## restriktive Layer

- Google Maps
- MSN Virtual Earth
- Yahoo! Maps
- Multimap

- Zoom / Pan
- Zoom Bar
- Mouse controls
- Layer Switcher (aka legend)
- Scale Ratio
- Scale Bar
- Permalink

# Was gibts noch?

- Markers
- Popups
- WMS-T
- Vektor integration (WFS-T)
- animated zooming and panning
- KML Layer



# Einbinden von WMS

```
<html>  
<head>  
<script src="http://openlayers.org/api/OpenLayers.js">  
</script>  
</head>  
<body>  
  
</body>  
</html>
```

# Einbinden von WMS

```
<html>
<head>
<script src="http://openlayers.org/api/OpenLayers.js">
</script>
</head>
<body>
  <div id="map"></div>

</body>
</html>
```

# Einbinden von WMS

```
<html>
<head>
<script src="http://openlayers.org/api/OpenLayers.js">
</script>
</head>
<body>
  <div id="map"></div>
  <script defer="defer" type="text/javascript">

  </script>
</body>
</html>
```

# Einbinden von WMS

```
<html>
<head>
<script src="http://openlayers.org/api/OpenLayers.js">
</script>
</head>
<body>
  <div id="map"></div>
  <script defer="defer" type="text/javascript">
    var map = new OpenLayers.Map('map');

    </script>
  </body>
</html>
```

# Einbinden von WMS

```
<html>
<head>
<script src="http://openlayers.org/api/OpenLayers.js">
</script>
</head>
<body>
  <div id="map"></div>
  <script defer="defer" type="text/javascript">
    var map = new OpenLayers.Map('map');
    var wms = new OpenLayers.Layer.WMS("OpenLayers WMS",
      "http://labs.metacarta.com/wms/vmap0", layers: 'basic');

  </script>
</body>
</html>
```

# Einbinden von WMS

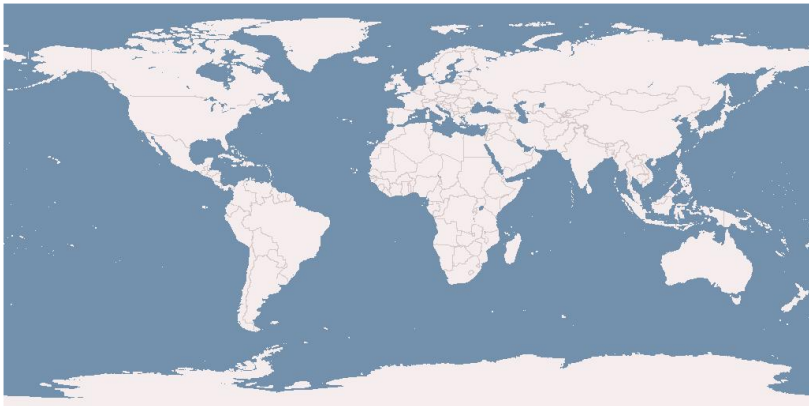
```
<html>
<head>
<script src="http://openlayers.org/api/OpenLayers.js">
</script>
</head>
<body>
  <div id="map"></div>
  <script defer="defer" type="text/javascript">
    var map = new OpenLayers.Map('map');
    var wms = new OpenLayers.Layer.WMS("OpenLayers WMS",
      "http://labs.metacarta.com/wms/vmap0", layers: 'basic');
    map.addLayer(wms);

  </script>
</body>
</html>
```

# Einbinden von WMS

```
<html>
<head>
<script src="http://openlayers.org/api/OpenLayers.js">
</script>
</head>
<body>
  <div id="map"></div>
  <script defer="defer" type="text/javascript">
    var map = new OpenLayers.Map('map');
    var wms = new OpenLayers.Layer.WMS("OpenLayers WMS",
      "http://labs.metacarta.com/wms/vmap0", layers: 'basic');
    map.addLayer(wms);
    map.zoomToMaxExtent();
  </script>
</body>
</html>
```

# Beispiel WMS



Live!?



# Übung: Erstellen einer Frida WMS OpenLayers Anwendung

- Binden Sie den zuvor erstellen Frida WMS in eine von Ihnen zu erstellenden minimalen OpenLayers-Anwendung ein.

# Einbinden von Google Maps

```
<html>
<head>
<script src="http://maps.google.com/maps?file=api">
</script>
<script src="http://openlayers.org/api/OpenLayers.js">
</script>
</head>
<body>
  <div id="map"></div>
  <script defer="defer" type="text/javascript">
    var map = new OpenLayers.Map('map');

    map.zoomToMaxExtent();
  </script>
</body>
</html>
```

# Einbinden von Google Maps

```
<html>
<head>
<script src="http://maps.google.com/maps?file=api">
</script>
<script src="http://openlayers.org/api/OpenLayers.js">
</script>
</head>
<body>
  <div id="map"></div>
  <script defer="defer" type="text/javascript">
    var map = new OpenLayers.Map('map');
    var layer = new OpenLayers.Layer.Google("Google");

    map.zoomToMaxExtent();
  </script>
</body>
</html>
```

# Einbinden von Google Maps

```
<html>
<head>
<script src="http://maps.google.com/maps?file=api">
</script>
<script src="http://openlayers.org/api/OpenLayers.js">
</script>
</head>
<body>
  <div id="map"></div>
  <script defer="defer" type="text/javascript">
    var map = new OpenLayers.Map('map');
    var layer = new OpenLayers.Layer.Google("Google");
    map.addLayer(layer);
    map.zoomToMaxExtent();
  </script>
</body>
</html>
```

# Beispiel Google Maps



Live?! Google Satellit

# Einbinden von Nasa WorldWind

```
<html>
<head>
<script src="http://openlayers.org/api/OpenLayers.js">
</script>
</head>
<body>
  <div id="map"></div>
  <script defer="defer" type="text/javascript">
    var map = new OpenLayers.Map('map', {'maxResolution': .28125,
                                          tileSize: new OpenLayers.Size(512, 512)});

    map.addLayer(layer);

  </script>
</body>
</html>
```

# Einbinden von Nasa WorldWind

```
<html>
<head>
<script src="http://openlayers.org/api/OpenLayers.js">
</script>
</head>
<body>
  <div id="map"></div>
  <script defer="defer" type="text/javascript">
    var map = new OpenLayers.Map('map', {'maxResolution': .28125,
                                          tileSize: new OpenLayers.Size(512, 512)});
    var layer = new OpenLayers.Layer.WorldWind("Bathy",
        "http://worldwind25.arc.nasa.gov/tile/tile.aspx?", 36, 4,
        T:"bmng.topo.bathy.200406");
    map.addLayer(layer);

  </script>
</body>
</html>
```

# Einbinden von Nasa WorldWind

```
<html>
<head>
<script src="http://openlayers.org/api/OpenLayers.js">
</script>
</head>
<body>
  <div id="map"></div>
  <script defer="defer" type="text/javascript">
    var map = new OpenLayers.Map('map', {'maxResolution': .28125,
                                          tileSize: new OpenLayers.Size(512, 512)});
    var layer = new OpenLayers.Layer.WorldWind("Bathy",
        "http://worldwind25.arc.nasa.gov/tile/tile.aspx?", 36, 4,
        T:"bmng.topo.bathy.200406");
    map.addLayer(layer);
    map.setCenter(new OpenLayers.LonLat(13.53460,52.43231),6);
  </script>
</body>
</html>
```



# Beispiel Nasa WorldWind



# Einbinden von Yahoo! Maps

```
<html>
<head>
<script src="http://api.maps.yahoo.com/ajaxymap?v=3.0&appid=euzuro-openlayers">
</script>
<script src="http://openlayers.org/api/OpenLayers.js">
</script>
</head>
<body>
  <div id="map"></div>
  <script defer="defer" type="text/javascript">
    var map = new OpenLayers.Map('map');

    map.zoomToMaxExtent();
  </script>
</body>
</html>
```

# Einbinden von Yahoo! Maps

```
<html>
<head>
<script src="http://api.maps.yahoo.com/ajaxymap?v=3.0&appid=euzuro-openlayers">
</script>
<script src="http://openlayers.org/api/OpenLayers.js">
</script>
</head>
<body>
  <div id="map"></div>
  <script defer="defer" type="text/javascript">
    var map = new OpenLayers.Map('map');
    var layer = new OpenLayers.Layer.Yahoo("Yahoo");

    map.zoomToMaxExtent();
  </script>
</body>
</html>
```

# Einbinden von Yahoo! Maps

```
<html>
<head>
<script src="http://api.maps.yahoo.com/ajaxymap?v=3.0&appid=euzuro-openlayers">
</script>
<script src="http://openlayers.org/api/OpenLayers.js">
</script>
</head>
<body>
  <div id="map"></div>
  <script defer="defer" type="text/javascript">
    var map = new OpenLayers.Map('map');
    var layer = new OpenLayers.Layer.Yahoo("Yahoo");
    map.addLayer(layer);
    map.zoomToMaxExtent();
  </script>
</body>
</html>
```

# Beispiel Yahoo! Maps



```
var controls = [  
    new OpenLayers.Control.PanZoomBar(),  
    new OpenLayers.Control.MouseDefaults(),  
    new OpenLayers.Control.Scale(),  
    new OpenLayers.Control.Permalink()  
];
```

```
var controls = [  
    new OpenLayers.Control.PanZoomBar(),  
    new OpenLayers.Control.MouseDefaults(),  
    new OpenLayers.Control.Scale(),  
    new OpenLayers.Control.Permalink()  
];  
  
var map = new OpenLayers.Map('map', controls: controls );
```

```
var controls = [  
    new OpenLayers.Control.PanZoomBar(),  
    new OpenLayers.Control.MouseDefaults(),  
    new OpenLayers.Control.Scale(),  
    new OpenLayers.Control.Permalink()  
];
```

```
var map = new OpenLayers.Map('map', controls: controls );
```

Beachten Sie die Verwendung von `controls`.



# Beispiel für eine benutzerdefinierte Kontrollsteuerung

