



Administration

Stand: 06.09.2023.2023
Client-Version 2.2.8

NETGIS GbR

Benediktinerstr. 32a
54292 Trier
Tel.: 0651-1704731
Fax: 0651-1704733
info@netgis.de
www.netgis.de

Inhalt

1. INSTALLATION UMN MAPSERVER UND POSTGRESQL/POSTGIS UNTER WINDOWS	2
1.1 STANDARD-VERZEICHNISSTRUKTUR UNTER WINDOWS	2
1.2 INSTALLATION PHP / UMN MAPSERVER	2
1.3 WEITERE EINSTELLUNGEN (PHP.INI, PROJ4)	3
1.4 INSTALLATION VON POSTGRESQL	4
1.5 INSTALLATION VON POSTGIS	4
1.6 OPTIMIERUNG DES DATENBANKSERVERS	5
1.7 POSTGRESQL ZUGANGSKONTROLLE	5
1.8 NETGIS-MAP-CLIENT INSTALLIEREN	6
2. ADMINISTRATION NETGIS-MAP-CLIENT	7
2.1 GRUNDSTRUKTUR UND AUFBAU DES CLIENTS	7
2.2 PARAMETERAUFRUF UND UNIVERSALSCHNITTSTELLE	7
2.3 VERZEICHNISSTRUKTUR IM WEB-VERZEICHNIS	9
2.4 ANPASSUNGEN IN DER ZENTRALEN KONFIGURATIONSDATEI BASECONFIG.PHP	11
2.5 SPRACHSTEUERUNG UND MEHRSPRACHIGKEIT	17
2.6 VERZEICHNISSTRUKTUR UND ANPASSUNGEN IM DATENVERZEICHNIS	19
2.7 ANPASSUNGEN IM DATENVERZEICHNIS: TEMPLATES	21
2.7.1 <i>Alkisbuch Autologin (mod_alkisbuch)</i>	22
2.7.2 <i>Edit-Templates (mod_edit)</i>	23
2.7.3 <i>Export-Modul Templates (mod_export)</i>	28
2.7.4 <i>Metadaten-Modul Templates (mod_metadata)</i>	29
2.7.5 <i>WMS / WFS Dienste-Modul Templates (mod_ogc)</i>	30
2.7.6 <i>Plan-Modul Templates (mod_plan)</i>	31
2.7.7 <i>Print-Templates</i>	33
2.7.8 <i>Query-Templates</i>	37
3. INFORMATIONEN ZU VERWENDETER OPENSOURCE SOFTWARE	41

1. Installation UMN Mapserver und PostgreSQL/PostGIS unter Windows

Diese Installationsanweisung bezieht sich auf die Verwendung eines Windows Betriebssystems mit dem Internet-Information Services (IIS). Die Verwendung eines Apache Webserver ist nicht Gegenstand dieser Dokumentation aber selbstverständlich auch möglich.

Da zum Betrieb des NETGIS-Map-Clients PHP und die Erweiterung PHP-Mapscript Verwendung findet muss diese Software ebenfalls auf dem System installiert werden. Eine Installation der CGI-Version ist optional und für den Betrieb des Clients nicht erforderlich.

Internet-Information Services (IIS) ist standardmäßig bei Windows 2012, 2016, 2019, und 2022 Server installiert, unter Windows 10 oder 11 muss der IIS eventuell nachinstalliert werden.

1.1 Standard-Verzeichnisstruktur unter Windows

- Standardinstallation des IIS in Verzeichnis **C:\Inetpub**;
- Standard-Root-Verzeichnis des Webserver: **C:\Inetpub\wwwroot**;
- Standard-CGI-Verzeichnis: **C:\Inetpub\wwwroot\cgi-bin**
Hinweis: Das cgi-bin Verzeichnis wird nur benötigt wenn UMN Mapserver als native CGI-Version auf dem Server laufen soll. Der Map-Client benötigt lediglich die PHP-Version.
- Temporäres Speicherverzeichnis Mapserver:
C:\Inetpub\wwwroot\~upload\tmp (volle Schreibrechte für IIS_User (Verzeichnissicherheit) auf Verzeichnis geben)
- Eventuell **tmp_loeschen.bat** zum löschen der temporären Dateien von Mapserver auf Server kopieren und als **Task** mindestens 1-mal täglich laufen lassen (häufiger ausführen lassen, wenn ein hoher Traffic zu erwarten ist).

1.2 Installation PHP / UMN Mapserver

Je nach Wunsch können unterschiedliche Versionen installiert werden.

Die Pakete sind für eine Installation unter C:/PHP vorkonfiguriert.

Abweichende Installationen bedürfen einer Anpassung der php.ini und Verknüpfungen im IIS. PHP muss im CGI-Modus installiert werden.

Kopieren Sie den Inhalt des gewählten Pakets einfach nach **C:/PHP**

Achtung: Häufig ist es nötig vor der PHP Installation die entsprechenden C++ Libraries über welche die PHP Version kompiliert wurde zu installieren. Dazu werden in der Regel Installer (c++lib/vcredist_x86.exe) mitgeliefert.

Konfiguration unter Windows 10, 11 und 2012, 2016, 2019, 2022 Server R2 (IIS 8+)

Wenn noch nicht vorhanden bzw. eingerichtet, muss zunächst der Rollendienst für CGI installiert werden:

Alle Programme/Verwaltung/Servermanager -> Webserver (IIS) -> Rollendienste hinzufügen -> Anwendungsentwicklung-> CGI

Danach muss PHP für den IIS konfiguriert werden:

- IIS Manager aufrufen (ausführen -> inetmgr) -> Handlerzuordnungen -> Moduluordnung hinzufügen
- Anforderungspfad: *.php Modul FastCgiModule Ausführbare Datei: C:\PHP\php-cgi.exe Name kann frei gewählt werden.
- Auf der Registerkarte Verben können optional noch Einschränkungen zu den http-Methoden gesetzt werden

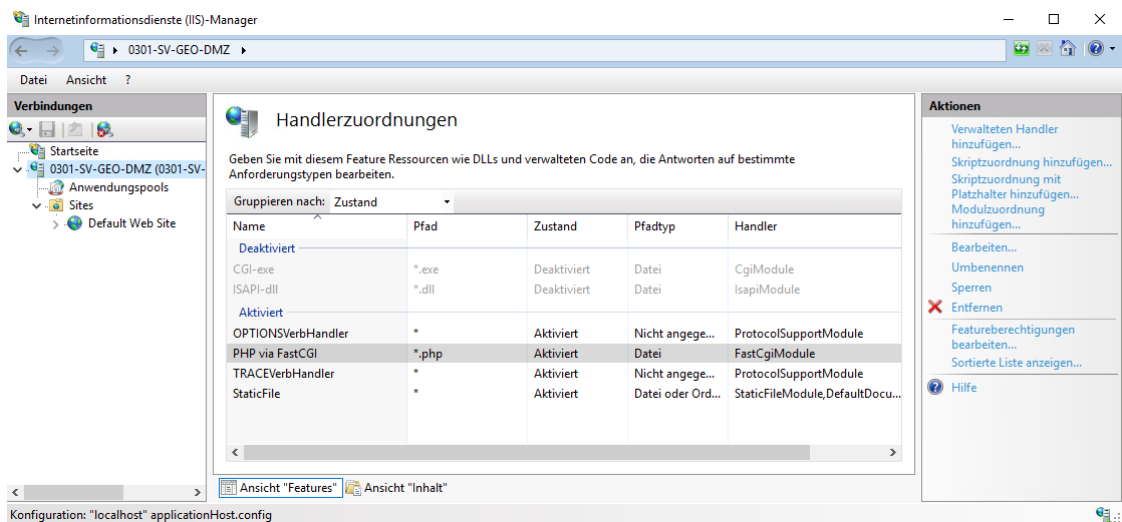


Abb. 1 Handlerzuordnungen im IIS-Manager unter Windows 2016 Server

1.3 Weitere Einstellungen (php.ini, proj4)

Folgende Einstellungen wurden in den vorkonfigurierten PHP-Paketen in der Datei php.ini bereits angepasst:

- Errorhandling: error_reporting = E_ALL & ~E_NOTICE
- Sessionpath : session.save_path= C:\PHP\session
- "Windows Extensions" : extension_dir = "C:\PHP\ext"
- unter Rubrik "Windows Extensions" wurden folgende dlls aktiviert:
extension=php_dbase.dll
extension=php_gd2.dll
extension=php_pgsqll.dll
extension=php_mapscript.dll

Für eine vollständige WMS- und PHP-Mapscript-Unterstützung muss auch die **Proj4 library** auf dem Server installiert werden.

Das Standardverzeichnis ist unter Windows das Verzeichnis „C:\Proj\nad“. Deshalb den Installations-Ordner **proj** nach **C:** kopieren (oder über Systemvariablen (PROJ_LIB) anderes Verzeichnis wählen)

Test PHP und PHP-Mapscript:

Zum Test von PHP erstellen Sie eine PHP-Datei mit folgendem Inhalt:

```
<?php phpinfo(); ?>
```

Zum Test der PHP Mapscript-Version:

```
<?php echo ms_GetVersion();?>
```

1.4 Installation von PostgreSQL

Grundsätzlich ist PostgreSQL nicht für den Betrieb des NETGIS-Map-Clients erforderlich. Allerdings setzen viele Module die Verwendung voraus, eine Installation ist deshalb zu empfehlen.

Die aktuellen Windows-Versionen (Installer und Binary-Pakete) für PostgreSQL stehen hier zum Download bereit: <http://www.postgresql.org/download/windows>
Die Windows Installationsfiles werden über die Firma EnterpriseDB bereitgestellt.

1. Installationsfile ausführen und Dialog folgen.
2. Installationsverzeichnis auswählen (in der Regel das Programmverzeichnis auf der C: Partition)
3. Datenverzeichnis auswählen, hier sollte am besten ein Verzeichnis auf der Datenpartition gewählt werden.
4. Passwort für den Superuser "postgres" vergeben, mit diesem Passwort melden Sie sich später auf dem Datenbankserver an.
5. Port vergeben auf dem der Datenbankdienst später laufen soll (in der Regel 5432)
6. default locale wählen
7. Installation für PostgreSQL ausführen und beenden.

1.5 Installation von PostGIS

1. Bei weiterer Installation aus dem PostgreSQL-Dialog kann an dieser Stelle POSTGIS auch über die Option „Launch Stack Builder“ über das Internet geladen werden.
Sie können den Installer für Postgis auch downloaden und direkt ausführen (Empfehlung bei der Installation auf Servern, die aus Sicherheitsgründen den Download über das Internet nicht zulassen)
2. Wählen Sie die Komponenten „Postgis“ und Create spatial database“
3. Im Dialog „choose install location“ empfiehlt sich das Standardverzeichnis unter PostgreSQL
4. Nach dem Dialog „databaseconnection“ bei dem Sie die Zugangsdaten des Datenbankservers eingeben und dem Dialog „databasename“ bei dem die Standarddatenbank für Postgis geändert werden kann ist die Installation abgeschlossen.

1.6 Optimierung des Datenbankservers

Die Voreinstellungen einer Standardinstallation von PostgreSQL sind für die Anwendung in einem Produktivsystem nicht ratsam da die Einstellungen eher für den parallelen Betrieb in einer Entwicklungsumgebung oder für Systeme mit schwacher Hardwareausstattung konzipiert wurden. Es sollten deshalb einige Anpassungen in der Konfiguration vorgenommen werden.

In diesem Zusammenhang empfehlen wir die Lektüre "PostgreSQL Administration" von Peter Eisenkraut aus dem O'Reilly Verlag.

Alle Parameter zur Optimierung können mit einem Editor in der zentralen Konfigurationsdatei **postgresql.conf** vorgenommen werden. Bei einer Standardinstallation befindet sich unter Windows dem Verzeichnis *C:\Program Files\PostgreSQL\9.3\data*. Die Beschreibung der optimalen Einstellungen der Parameter würde den Rahmen des Tutorials sprengen, es sollen deshalb lediglich die Parameter aufgelistet werden, die vor einer Inbetriebnahme angepasst oder überprüft werden sollten.

Einstellungen zur Verbindungskontrolle:

- `listen_addresses` (vgl. Kap. Zugangskontrolle)
- `max_connections` (maximale Anzahl von Verbindungen die eine Instanz des DB-Servers gleichzeitig offen haben kann)
- `ssl` (aktiviert die SSL-Unterstützung des DB-Servers).

Einstellungen zur Speicherverwaltung:

- `shared_buffers` (legt die Größe des Shared-Buffer-Pools einer DB-Instanz fest)
- `work_mem` (Obergrenze des zur Verfügung stehenden Arbeitsspeichers für DB-Operationen wie Sortieren oder Verknüpfungsalgorithmen)
- `maintenance_work_mem` (Obergrenze des zur Verfügung stehenden Arbeitsspeichers für Verwaltungsoperationen zur Erzeugung oder Änderung von DB-Objekten)
- `max_fsm_pages`

Einstellungen zur Wartung, Planeinstellungen und Logging:

- `wal_buffers`
- `checkpoint_segments`
- `checkpoint_timeout`
- `effective_cache_size`
- `log_line_prefix`

1.7 PostgreSQL Zugangskontrolle

Nach der Installation ist die Datenbank lediglich vom lokalen Server erreichbar. Zur Freigabe der Datenbank in einem Netzwerk für andere Hosts müssen folgende Einstellungen vorgenommen werden (einfaches Beispiel):

postgresql.conf

Hier muß der Parameter „`listen_addresses`“ angepasst werden. Der Parameter konfiguriert die IP-Adressen auf denen Datenbankverbindungen entgegen genommen werden. Es können IP-Adressen oder Hostnamen eingestellt werden.

Normalerweise wird die Voreinstellung so geändert, dass Verbindungen von allen Adressen (Platzhalter *) möglich sind:

```
listen_addresses = '*'
```

Andere Einstellungen siehe Dokumentation.

pg_hba.conf

#	TYPE	DATABASE	USER	CIDR-ADDRESS	METHOD
	host	all	all	192.168.178.8/32	md5

In diesem Fall erhält ein einzelner Host über Passwortabfrage die Berechtigung für alle Datenbanken und User, andere Einstellungen siehe Doku.

Bitte beachten Sie die entsprechende Portfreigabe in der Firewall des Servers!

Außerdem sollte nach den Änderungen der PostgreSQL-Dienst neu gestartet werden.

1.8 NETGIS-Map-Client installieren

- Installations-Ordner **servicename** in der Regel **C:\Inetpub\wwwroot** kopieren
Installations-Ordner **data_servicename** in z.B. **D:\data_servicename** kopieren
- Client Datenbank auf dem PostgreSQL erzeugen und Backup importieren
- Berechtigungen (Windows-Sicherheit) setzen: Schreibzugriff für den IIS-User auf folgende Verzeichnisse:

- D:\ data_servicename

(Pfad kann variieren, vgl. Einstellungen in der baseconfig.php)

- Start des Clients: <http://servername/servicename/index.php>

2. Administration NETGIS-Map-Client

Der Client ist auf 2 Verzeichnisse verteilt:

das Web-Verzeichnis: Bei der Namensgebung wird in der Regel auf folgende Nomenklatur Wert gelegt: **servicename**

Das Daten-Verzeichnis: Bei der Namensgebung wird in der Regel auf folgende Nomenklatur Wert gelegt **data_servicename**

2.1 Grundstruktur und Aufbau des Clients

Im Web- Verzeichnis liegen alle HTML- CSS-, PHP- und Javascript-Dateien, auf die über das Web direkt zugegriffen werden kann.

Grundsätzlich ist der Client One-Page-Aplication konzipiert. Die Startseite heißt **index.php** und liegt im root-Verzeichnis des Clients.

Die Datei **login.php** im root-Verzeichnis dient dem Login für den Zugriff auf geschützte Dienste oder der Clientadministration, **logout.php** dient dem Beenden der jeweiligen Sitzung.

Die Datei **print.php** dient dem einfachen HTML-Druck und wird durch einen entsprechenden Parameterruf aus der index.php angesteuert.

version.php beinhaltet Informationen zur aktuellen Version und umfangreiche Hinweise bzgl. der Änderungen zu vorherigen Versionsständen.

2.2 Parameterruf und Universalschnittstelle

Bei einem Aufruf des Clients über die Datei index.php können wie beim Linkgenerator erzeugt folgende Variablen in der URL mitgegeben werden:

Beispiel:

`http://localhost/clientname/index.php?lang=de&zl=10&x=3461360&y=5483508
&bl=stadtplan_grau&bo=1&lo=1&layers=layer1,layer2&service=servicename1`

Folgende Variablen sind dabei möglich:

lang = Sprache (de,en,fr)

zl = Zoomlevel, Ganzzahl ,hängt von der Anzahl der Zoomstufen ab.

x = der Rechtswert bzw. die X-Koordinate (in der Regel metrische Koordinaten)

y = der Hochwert bzw. die Y-Koordinate (in der Regel metrische Koordinaten)

bl = aktive Hintergrundkarte als Ebenenname

bo = Transparenz der Hintergrundkarte (Zahl zwischen 0 und 1)

lo = Transparenz der Overlays durch komma separiert (Zahlen zwischen 0 und 1)

layers = Namen der Overlay-Ebenen Komma separiert

service = Name des Zielservices

Eine Universalschnittstelle wird über die Datei **mapinterface.php** bereitgestellt. Die Schnittstelle funktioniert für den Desktopclient wie auch für die mobile Variante des jeweiligen Services. Das Ziel wird über das Dropdown Feld "client" eingestellt. Es können beliebige Objekte einer abfragbaren Ebene (qlayer) über

Angabe des Id-Feldes (qfield) und kommaseparierte Ids (qidstring) an die Anwendung weitergegeben werden. Zusätzlich muss der Datentyp des Feldes (qidtyp) angegeben werden, ein Buffer für den Zoombereich (qbuffer) ist optional.

Weitere Variablen können wie im Parameterruf beschrieben eingetragen werden.

Die Einzelparameter können über [clienturl]/**mapinterface.php** getestet und ein entsprechender Link generiert werden.

NETGIS-Mapinterface Testform (* Pflichtvariablen)		
Variable	Testfeld	Erläuterung
client	Desktop ▾	Ziel-Client
qlayer *	<input type="text"/>	Abfragelayer, Komma separiert
qfield *	<input type="text"/>	Abfragefeld
qidtyp *	text ▾	Datentyp der Objektkids
qid *	<input type="text"/>	Objektkids, Komma separiert bei Text Case Sensitive!
layers	<input type="text"/>	Layerzusammenstellung, Komma separiert Um die Objekte zu hiliten muss der jeweilige Abfragelayer mit angegeben werden!
qbuffer	<input type="text"/>	Bufferabstand im Kartenfenster um Objekt(e) in Meter Standard = 0 (optional)
bl	<input type="text"/>	Hintergrundkarte Standard in Basiskonfiguration (optional)
zoomobject	true ▾	Soll auf Objekt(e) gezoomt werden? true / false, Standard = true (optional)
editobject	false ▾	Editiermodus (true / false, Standard = false) (optional)
editgroup	<input type="text"/>	Name der User Gruppe, die eingeloggt sein muss, damit editobjekt genutzt werden kann (optional)
service	<input type="text"/>	Servicename (optional)
Werte an Schnittstelle übergeben		
Generierter Link:		
http://localhost/kartendienste_sgdn/mapinterface.php?qlayer=&qfield=&qidtyp=text&qid=&client=&zoomobject=true		
109 Zeichen		
Hinweis: Die maximale Zeichenlänge der von Browsern verarbeitbaren URL beträgt 2048 Zeichen.		

Abb. 2 Testformular der Universalschnittstelle des Map-Clients

2.3 Verzeichnisstruktur im Web-Verzeichnis

Im Folgenden werden die Unterordner des Web-Verzeichnisses genauer betrachtet. Dabei sind **rot markierte Variablen** beim Anlegen eines **neuen** Services zwingend anzupassen.

admin /

Enthält alle relevanten php-Dateien für die Clientverwaltung bzw. Ebenen-Administration sowie die Benutzerverwaltung. Weitere Informationen finden Sie im Kapitel *Ebenen-Administration*.

ajax/

Alle als Ajax-Funktion aufgerufenen PHP-Skripte des Clients. Ajaxfunktionen werden grundsätzlich über das Javascriptframework *jQuery* aufgerufen.

config/

Alle Konfigurationsdateien, die einer Parametrisierung des Clients dienen. Folgende Dateien werden hier genauer betrachtet:

config/img/

Dieses Verzeichnis dient der Speicherung von Grafiken für die Legendenausgabe (im Unterverzeichnis "symbol") sowie Grafiken für den Headerbereich.

Da die Daten über eine Upload-Funktion in das Verzeichnis geladen werden können, muss eine Schreibberechtigung für den IIS-User auf dem Verzeichnis liegen.

config/print/

Dieses Verzeichnis dient der Speicherung von Grafiken die in die Druck-Templates eingebunden werden. Dazu gehören auch die Vorschaubilder der Seitenlayouts.

config/upload/

Dieser Ordner speichert Dokumente und Grafiken, die über den Client verlinkt wurden bzw. erreichbar sein sollen. Da die Daten über eine Upload-Funktion in das Verzeichnis geladen werden können, muss eine Schreibberechtigung für den IIS-User auf dem Verzeichnis liegen.

1_ngms_global.js

Diese Datei speichert die globalen Variablen zur Anpassung eines Clients. Neben den Zoomstufen bzw. Scales können hier auch die Projektion und die Layoutmasse konfiguriert werden.

baseconfig.php

In dieser Datei werden die wesentlichen Konfigurationsvariablen des Clients gespeichert. Eine detaillierte Beschreibung der Variablen finden Sie im nächsten Kapitel dieser Dokumentation.

connections/

Dieses Verzeichnis beinhaltet die Datenbankverbindungsdaten:

db_connection.php

in dieser Datei beinhaltet die Verbindungsdaten zur Clientdatenbank unter PostgreSQL. Folgende Variablen sind dabei relevant:

```
$dbHost = "localhost";  
$dbPort = "5093";  
$dbDatabase = "ziel_client";  
$dbUser = "postgres";  
$dbPw = "streng#geheim";
```

css/

Cascading Style Sheets des Clients.

img/

Alle Bilddateien des Clients. Im Unterordner menu liegen die verwendeten Button-Bilder des Menüs.

includes/

Die inkludierten php-Dateien des Clients sind hier abgelegt. Dabei handelt es sich um Funktionen und, die in anderen Dateien eingebunden sind und mehrfach benutzt werden.

javascript/

hier sind die im Client verwendeten Javascript-Dateien abgelegt. Im Unterordner dev liegen die für die Entwicklung relevanten Sourcen. Diese werden über einen Build-Prozess in komprimierten Zieldateien ngms_event.min.js und ngms_mod.min.js abgelegt. In der Basiskonfiguration (\$devmode bzw. \$devmode_mod) kann darüber entschieden werden, ob die Developer Sourcen genutzt werden oder die Komprimierten.

mod_xy/ (Optional)

Das Clientsystem ist modular aufgebaut. In den Ordnern mit dem Kürzel **mod_** am Anfang befinden sich alle für die jeweiligen Module relevanten Sourcen. Eine nähere Beschreibung dieser Verzeichnisse ist nicht vorgesehen.

2.4 Anpassungen in der zentralen Konfigurationsdatei baseconfig.php

In der Datei „config/baseconfig.php“ werden die wesentlichen Konfigurationsvariablen des Clients gespeichert. Diese werden im Folgenden genauer beschrieben:

\$maintenancemode = false;

Wenn der Client in den Wartungsmodus aktiviert werden soll auf true (false/true)

\$devmode = false;

Ist der Wert auf true gestellt werden die unkomprimierten Javascript Dateien der Basisanwendung für ein besseres Debugging eingebunden, Developer Mode

\$devmode_mod = false;

Ist der Wert auf true gestellt werden die unkomprimierten Javascript Dateien der Module für ein besseres Debugging eingebunden, Developer Mode

\$hostname = **\$_SERVER["HTTP_HOST"]**;

\$protocol = "http:";

if(isset(\$_SERVER["HTTPS"])){if(\$_SERVER["HTTPS"] == "on"){ \$protocol = "https:";} }

Host (Servername) der Clientapplikation, wird dynamisch ausgelesen, deshalb sollte normalerweise diese Variablen nicht geändert werden. Protokoll der Clientapplikation, wird dynamisch ausgelesen, deshalb nicht ändern. Zulässige Protokolle: *http*, *https*

\$usesecurelogin = false;

Bei true: Nur das Login (Passwortübertragung) über eine sichere https Verbindung vornehmen (true, false). Die erzeugten Links aus dem login laufen wiederum unverschlüsselt über http

\$hide_loginbutton = false;

Login Button im Client nicht darstellen (per default false)

\$hide_clientadmin = false;

Link zum Clientadmin u. Service im Logindialog nicht darstellen (per default false)

\$stop_ie = false;

Internet Explorer verbieten

\$login_message = "";

Individueller Hinweistext für den Login Dialog (kann beliebigen HTML Code enthalten)

\$servicename = " ***servicename***";

definiert den Default Servicennamen des Clients und muss beim Anlegen eines neuen Dienstes angepasst werden.

\$sessionname= **\$servicename**;

Sessionprefix für alle Sessionvariablen in der Clientapplication (als Standard der default Servicename).

\$service_srid = " 25832";

Der Service EPSG Code für das verwendete Koordinatensystem.

\$service_title = "NETGIS Client";

Titel des default Service / Admin Bereich

\$servicenamemobile = "netgis_client/mod_mobile";
relativer Pfad der mobilen Clientapplication (optional nur bei aktiver mobilen Version)

\$applicationurl = \$protocol."/".\$hostname."/".\$servicename;
Applikationsvariable der Anwendung, wird dynamisch erzeugt, nicht ändern

**\$applicationurlmobile =
\$protocol."/".\$hostname."/".\$servicenamemobile;**
Mobile Application URL (nicht ändern!)

```
if (function_exists('msGetVersionInt')) {  
    $mapscriptSwig = true;  
}else{  
    $mapscriptSwig = false;  
}
```

In der Version 2.1 ist diese Variable hinzugekommen und wird in der Regel über eine automatische Überprüfung der PHP-Mapscriptversion gesetzt, nicht ändern.

define("MAPFILE_PATH", "C:/Inetpub/db/data_netgis_client/");

Beispiel unter Windows

define("MAPFILE_PATH", "/var/www/db/data_netgis_client/");

Beispiel unter Linux

definiert den physischen Pfad zum Datenverzeichnis mit Mapfiles and Maptemplates (Pfad mit Slash auch am Ende!) Muss beim Anlegen eines neuen Dienstes angepasst werden.

define("MS_TMP_PATH", "C:/inetpub/wwwroot/~upload/tmp/");

Pfad des temporäre Verzeichnisses der Mapserver Dateien (Pfad mit Slash auch am Ende!)

define("MS_TMP_URL", "/~upload/tmp/");

temporäre Verzeichnis-URL (Url mit Slash auch am Ende!)

\$rquerylayer = "mydhm";

Name der Mapserver-Ebene für eine Höhenabfrage

\$baselayeropacity = 100;

Anfangstransparenz der Hintergrundkarte in %

\$activebaselayer = "stadtplan_grau";

aktive Default-Hintergrundkarte falls in der Clientadministration kein anderer Wert eingetragen ist (Servicebereich).

\$sortlayer = "pos";

Sortierung des Layerbaums alphabetisch (auto) oder nach Feld pos in der Tabelle Layer der Clientdatenbank.

\$placeSearchNoAddress = false;

Handelt es sich um eine reine Ortssuche ohne Straßen und Hausnummern? (true, false ist Standard)

\$hsnr_mannheim_model = false;

Mannheimer Sortierungsmodell für Hausnummern

\$placeSearchautocomplete = true;

Ortssuche mit autocomplete oder nicht (true, false) Hinweis bei streetsearch auf true setzen, bei google auf false!

\$placeSearchNoAdress = false;

Handelt es sich um eine reine Ortssuche ohne Straßen und Hausnummern? (true, false ist Standard)

\$osmrequestemail = "info@netgis.de";

Authentifizierungs E-Mail für OSM Anfragen (Bitte Standardmailadresse anpassen)

\$mylangs = array("de","en", "fr");

Sprachversionen die bei Aktivierung im service (erster Eintrag ist Standard)

\$multilangstyle = "image";

Soll ein Sprachlink als Textlink oder als Bild (Flagge) ausgegeben werden?
mögliche Variable text,image

\$printcopy = "(C) Copyright Stadt xy";

Copyright für HTML-Druckausgabe

\$copyinmap = 'Stadt xy © 2014';

Copyrightvermerk im Kartenfenster, kann auch HTML Elemente beinhalten.

\$lefttab_active = "layer";

welches Modul soll im linken Tab geladen werden (default "layer", dann wird der Ebenenbaum geladen), kann auch über die Clientadministration im Servicebereich eingestellt werden.

\$queryfirstpanelopen = true;

Soll bei Abfragen das erste Pannel geöffnet sein? (true, false))

\$querybufferdist = 40;

Zoom-Bufferdistanz um Objekte nach Abfrage

\$queryencoding = "UTF-8";

Standard Encoding der query Tabellen, in der Regel UTF-8, bei Shapefiles meist LATIN1 (iso-8859-1).

\$legendencoding = "UTF-8";

Standard Encoding des Legendens Templates (iso-8859-1 oder utf-8)

\$zoombox_tool = false;

Werkzeug Zoomboxtool im Menü aktivieren.

\$client_hilfe = true;

Soll die Client Hilfe im Menü aktiviert werden? (true / false)

\$client_hilfe_url = "http://www.netgis.de/clienthilfe/pages/ ";

URL zu Hilfe (absolut oder relativ ohne Dateinamen)

\$gpsmodul = true;

GPS Modul im Client aktivieren / deaktivieren (im Werkzeug Menü)

\$redlining_hide = true;

Setzt fest, ob Redlining Ordner unsichtbar gemacht wird für nicht eingeloggte User. Update Infos und Sourcen in mod_edit_update_redlining

\$redlinefolder = "Redlining";

Name des Redlining Ordners in dem sich alle Redlining Ebenen befinden

\$logolinkurl = "http://www.netgis.de";

Soll ein Link für das obere Logo-Banner erzeugt werden vollständige URL eintragen (Leer lassen wenn kein Link, vgl. CSS ID logolink in Service Banner CSS)

\$logolinktitle = "Homepage von NETGIS";

titel-Tag zur Variablen \$logolinkurl

Mapserver-Modul (Basismodul, in der Regel aktiviert.)

\$mapserver_layer_norender = array("agro_rlp_query");

Layernamen die in map.php nicht gezeichnet werden sollen, die Abfrage funktioniert normal

\$mapserver_layer_alwaysrender = array("maske","credits");

Layernamen die in map.php immer aktiv sein sollen gezeichnet werden sollen

\$mapserver_layer_nomultiquery = array("agro_rlp_query");

Layernamen die in query.php keine Multiquery zulassen sollen (Performance Schutz)

\$wms_write_mapfile = true;

Automatisiertes Schreiben von WMS Mapfiles

\$wms_proxy = false;

Soll ein Proxy für die serverseitige Kontaktierung von WMS aktiviert werden (true oder false)

\$wms_proxy_host = "proxy";

falls \$wms_proxy auf true wird hier der proxy Host eingetragen

\$wms_proxy_port = 8080;

falls \$wms_proxy auf true wird hier der proxy Port eingetragen

\$wms_proxy_auth = false;

Soll eine Authentifizierung für den Proxy vorgenommen werden (true oder false)

\$wms_proxy_user = "myuser";

Username für die Proxyauthentifizierung

\$wms_proxy_pwd = "mypwd";

Passwort für die Proxyauthentifizierung

\$custom_maptilesize = 256;

Maptilesize global ändern (mus integer sein, min 256, max 1024), default ist 256

Folgende Variablen dienen zur Aktivierung bzw. Einstellung des OGC und Metadaten Moduls:

\$mod_ogc = true;

OGC-Modul (WMS WFS Client) aktivieren

\$mod_ogc_cgimode = false;

Soll die Mapserver FastCgi Variante genutzt werden oder die PHP Mapscript (true oder false, default false)

\$mod_cgi_mapservurl = \$protocol."/".\$hostname.'/ms/mapserver.fcgi?';
CGI Ziel-URL zur Mapserver-bin wenn \$mod_ogc_cgimode aktiv

\$mod_metadata = true;
Metadaten Client aktivieren

Seit der Version 2.6 ist ein umfangreiches **3D Modul** / Client in dem Framework aktivierbar, die Variablen in der baseconfig werden hier definiert:

\$mod_3d = true;
3D Modul aktivieren, nur wenn diese Variable auf true steht, werden die eD Werkzeuge (service und Layer) im Client Admin sichtbar

\$mod_3d_iontoken= "xyzmeintokenkey";
Falls benötigt kann hier der Cesium ION Token eingetragen werden

\$mod_3d_folderoptions = array(1378);
Folder die als Optionsauswahl im Layerbaum ausgegeben werden sollen (z.B. Hintergrundkarten), Array der Folderids (vgl. Admin Bereich)

\$mod_3d_clipbox_color = array(255,255,0);
Clipboxfarbe (RGB Array)

\$mod_3d_clipbox_width = 2;
Clipbox Linienstärke(in px)

\$mod_3d_searchmode = 'streetsearch';
Suchmodus für 3D Client (Mögliche Varianten: streetsearch,rlp)

\$mod_3d_camera_smooth = true;
Soll der Panmode bei allen Services smooth gestellt sein (eventuell Performance)

\$mod_3d_intro_confirm_label = "Nutzungsbedingungen verstanden";
Info Text neben Checkbox bei Confirm-Modal Fenster

\$mod_3d_copyinmap = 'Text oder HTML';
Copyrightinfo in jedem service des 3D Client als HTML oder Text

\$mod_3d_banner_image = "../config/img/logo_3d.png";
Banner Logo mit relativem Pfad bzw. URL

\$mod_3d_banner_image_hover = "../config/img/logo_3d_hover.png";
hover Banner Logo mit relativem Pfad bzw. URL

Seit Version 2.8 steht im Admin Bereich ein File Manager zur Verfügung

\$tinyfilemanager = true;

File Manager unter Tools aktivieren (true) oder deaktivieren (false)

\$tinyfilemanager_auth_users_array = array(
 'admin' => '\$2x10\$HABGqwWztb1MHifZ3cGTxEINF,
 'test' => '\$2y\$10\$KM5ijJYLj7CTftlc9ngg2,
 'netgis' => '\$2y\$4410\$HABGqM5ijJYMHifZ3cGT2e8x
);

User mit definiertem Passworthash, Hash für jeden User mit dieser PHP Funktion erzeugen: password_hash('password here', PASSWORD_DEFAULT)

\$tinyfilemanager_root_path = "c:/inetpub";

Pfad des Root Verzeichnisses für den Filemanager (Pfade ohne Slash am Ende!)

\$tinyfilemanager_directories_users_array = array(
 'test' => "c:/inetpub/wwwroot/~upload"
);

Es kann hiermit für die User einen Rootpath definieren falls der vom Root verzeichnis abweichen soll (Pfade ohne Slash am Ende!)

Zusätzlich gibt es noch zahlreiche optionale **Modulvariablen**, die nur bei Aktivierung der Module nötig und nicht Bestandteil dieser Dokumentation sind. Die Variablen werden jeweils mit Kommentaren in der baseconfig.php erklärt.

2.5 Sprachsteuerung und Mehrsprachigkeit

Alle Texte im Client (teilweise auch in Basismodulen) sind parametrisiert und in Konfigurationsdateien gespeichert.

In dem Verzeichnis **includes/language/** sind diese Konfigurationsdateien abgelegt und so für jeden Client individuell einstellbar.

Zum Zeitpunkt sind folgende Sprachen verfügbar:

- Deutsch(de)
- Englisch(en)
- Französisch(fr)

Die Mehrsprachigkeit bezieht sich lediglich auf das Frontend, der Administratorbereich und das Editiermodul bleiben deutsch.

Der Aufruf einer anderen Sprache läuft über die GET oder POST-Variable "lang", z.B. index.php?lang=en

Bei keinem Parameter wird automatisch die Standardsprache aufgerufen.

Die Mehrsprachigkeit wird servicebasiert über die Clientadministration eingestellt.

Die Sprachkonfiguration wird in der Datei **check_lang.php** vorgenommen.

Folgende Grundfunktionen können eingestellt werden:

Als Basis für die Einstellung der Sprachversionen dient ein Array (**\$mylangs** aus der baseconfig.php). Die Reigenfolge entspricht der Reihenfolge der Aliaseingabe für die Ebenen bzw. Hintergrundkartenzusammenstellungen in der Clientadministration. Soll ein Sprachlink als Text („text“) oder Bilde („image“) ausgegeben werden, kann dies über die Variable **\$multilangstyle** in der baseconfig.php eingestellt werden. Als Bild werden die Fahnnensymbole (z.B. de.gif) ausgegeben.

Folgende Objekte können bei einer Mehrsprachigkeit in den XML-Templates eingestellt werden (vgl. Anpassungen im Datenverzeichnis: Templates):

a) Print-Templates (z.B. template/print a4_landscape.xml):

Objekte "name" und "pretext" (textcell und varctextcell) mit jeweiligen Sprachzusatz "_fr" bzw "_en". Standardsprache unverändert.

b) Legenden-Template: (template/legend_template.html)

Die Standardsprache hat den Namen ohne Zusatz: "legend_template.html" Die anderen Sprachen "legend_template_en.html" bzw. "legend_template_fr.html" usw.

c) Query-Templates (z.B. template/query/layerxy.xml):
Objekte "alias" und "linkalias" mit jeweiligen Sprachzusatz "_fr" bzw "_en".
Standardsprache unverändert.

z.B.:

```
<field>
  <name>file_name</name>
  <alias>Gesetz</alias>
  <alias_en>Law</alias_en>
  <alias_fr>Droit</alias_fr>
  <link>true</link>
  <linkprestring>gesetzte/</linkprestring>
  <linkpoststring>.html</linkpoststring>
  <linkalias>Verweis auf Dokument</linkalias>
  <linkalias_en>Reference to document</linkalias_en>
  <linkalias_fr>Référence au document</linkalias_fr>
</field>
```

2.6 Verzeichnisstruktur und Anpassungen im Datenverzeichnis

Standardverzeichnis (IIS, Windows): *C:\Inetpub\db\data_servicename*

Das Datenverzeichnis ist außerhalb des Webs abgelegt, um direkte Zugriffe über das Internet zu verhindern (z.B. auf die Geodaten). Deshalb sollte auch kein virtuelles Verzeichnis im Webserver hierauf angelegt werden.

Folgende Verzeichnisse und Dateien sind bei der Anpassung des Clients bzw. beim Neuanlegen eines Services von Relevanz:

map.map

die Mapdatei ist die zentrale Steuerungsdatei für die Konfiguration des Kartendienstes mit UMN Mapserver. Alle Fachdaten und Grundeinstellungen werden hier definiert. Für einen neuen Kartendienstes muss diese Datei angepasst bzw. neu angelegt werden. In der Regel werden alle im Mapserver definierten Ebenen nach Thema gruppiert in einer .map-Datei konfiguriert.

Eine umfangreiche Dokumentation finden Sie hier:

<http://www.mapserver.org/mapfile/>

basis.map (optional)

Um bei umfangreichen Projekten die Map-Datei auf die Fachdaten zu reduzieren, werden die Basisdaten (z.B. Topografische Karten oder Grenzen) in dieser Map-Datei gespeichert. Die Datei wird wiederum in der Datei map.map inkludiert:

```
INCLUDE 'basis.map'
```

hilite.map

In dieser Map-Datei werden für die Visualisierung von Abfrageergebnissen für Punkt, Linien und Polygone definiert. Die Farben können entsprechend angepasst werden. Die Datei wird in der Datei map.map inkludiert:

```
INCLUDE 'hilite.map'
```

wms_baselayer.map

Alle über das Admintool definierten WMS Dienste müssen auch für den Mapserver definiert werden da sonst kein Druck der Daten möglich wäre. In diese Datei werden alle als Hintergrundkarten Ebenen verknüpften WMS Dienste automatisch vom System generiert. Eine Bearbeitung der Datei sollte deshalb nicht erfolgen!

Die Datei wird wiederum in der Datei map.map inkludiert und muss deshalb vorhanden sein:

```
INCLUDE wms_baselayer.map'
```

wms_overlay.map

In diese Datei werden alle als Overlay-Ebenen verknüpften WMS Dienste automatisch vom System generiert. Eine Bearbeitung der Datei sollte deshalb nicht erfolgen! Die Datei wird wiederum in der Datei map.map inkludiert und muss deshalb vorhanden sein:

```
INCLUDE wms_baselayer.map'
```

fonts/

in diesem Ordner sind die verwendeten Symboldateien und Truetype-Schriftarten abgelegt. In der Datei symbols.sym sind mapserverspezifische Symbole definiert. Wenn Sie eine neue Schriftart für Ihre map-Definition verwenden wollen müssen

diese nach dem Kopieren in den Ordner auch in der fonts.list als neue Zeile aufgeführt werden.

geodata/

in diesem Verzeichnis sollten die verwendeten Geodaten abgelegt werden. Natürlich kann jedes andere Verzeichnis auf dem Server Verwendung finden, allerdings muss dann vorher der entsprechende datenpfad in der .map-Datei angepasst werden (z.B. SHAPEPATH "geodata") oder jeweils der vollständige Datenpfad für jeden Layer.

reference

Im Ordner Reference liegt die Übersichtskartenvorlage (im Rasterformat), die in der .map-Datei definiert wurde. Bei einem Austausch dieser Datei müssen die entsprechenden Parameter in der .map-Datei angepasst werden. Diese Dateien sind lediglich bei einer Ausgabe im PDF-Druck von relevanz.

2.7 Anpassungen im Datenverzeichnis: Templates

template

In diesem Ordner werden template-Dateien für den Kartendienst abgelegt. Folgende Dateien sind in dem Verzeichnis abgelegt:

legend_template.html

dient der Anpassung einer dynamisch erzeugten HTML-Legende. Legendensymbole die nicht von Mapserver generiert werden müssen das Bildformat .png oder .jpg haben! Bei der Anpassung können alle Elemente der der CGI-Version Anwendung finden. Eine umfangreiche Dokumentation finden Sie hier:

http://mapserver.org/de/output/html_legend.html

query.html

dient als Platzhalter für die Query-Funktion. Die Abfragen werden im Client über php-mapscript realisiert, deshalb ist die Datei leer und sollte unverändert bleiben.

spatial_query.xml (optional)

Um für das Exportmodul eine räumliche Verschneidung der zu exportierenden Daten vornehmen zu können dient diese Konfigurationsdatei.

Folgende XML-Objekte sind dabei einstellbar:

data = root-Objekt der XML-Datei (Wurzelement), umschließt alle anderen Objekte und darf nur einmal definiert werden

layer = Layer-Objekt, beschreibt jeweils einen Layer im Mapfile und kann beliebig oft definiert werden. Dieses Objekt umschließt die folgenden:

name = Name der Ebene (Pflichtelement)

alias = Alias für die Bezeichnung der Ebene (Pflichtelement)

dbcon = Name der Datenbankverbindungsdatei (ohne .php) im Ordner clientname/connections (Pflichtelement)

table = Name Datenbanktabelle / View (Pflichtelement)

searchfield = Name des Autocomplete-Suchfeldes (nur Strings!) (Pflichtelement)

id = Name des Feldes mit dem Primärschlüssel (Pflichtelement)

geom = Name des Geometriefeldes (Pflichtelement)

In den folgenden Unterordnern liegen XML-Templates zur Konfiguration Planwerken, Abfragen, Editierlayern und Druckvorlagen

2.7.1 Alkisbuch Autologin (mod_alkisbuch)

template/alkisbuch

alkisbuch_list.xml

Mit diesem XML-File können Sie ein Autologin in das Topographics ALKIS Buch für verschiedene Services konfigurieren.

Folgende XML-Objekte sind dabei einstellbar:

data = root-Objekt der XML-Datei (Wurzelement), umschließt alle anderen Objekte und darf nur einmal definiert werden

service = Service-Objekt, beschreibt jeweils einen Service und kann beliebig oft definiert werden. Dieses Objekt umschließt die folgenden:

name = Usergruppe aus der Clientanmeldung

alkisuser = Name des Users für die Anmeldung im ALKISBuch

alkispw = Passwort des Users für die Anmeldung im ALKISBuch

Beispiel eines Templates:

```
<?xml version="1.0" encoding="UTF-8" ?>
<data>
  <service>
    <name>user_group_x</name>
    <alkisuser>user_x</alkisuser>
    <alkispw>pw_x</alkispw>
  </service>
</data>
```

2.7.2 Edit-Templates (mod_edit)

template/edit

ebenennamen.xml

in diesem Verzeichnis ist für jede editierbare Ebene aus dem Modul mod_edit eine XML-Konfigurationsdatei für die Eingabemasken abgelegt. Die Daten der jeweiligen Ebene müssen zwingend aus einer PostGIS-Datenbank kommen. Die Bearbeitung von Sichten (Views) ist nicht oder nur mit entsprechenden Regeln möglich.

Um ein Edit Template verwenden zu können, benötigen Sie auf jeden Fall für die Zielebene auch ein Query Template mit einem dbinfo Objekt (vgl. Kapitel Query-Templates).

Folgende XML-Objekte sind dabei einstellbar:

daten = root-Objekt der XML-Datei (Wurzelement), umschließt alle anderen Objekte und darf nur einmal definiert werden

refreshlayer = optionales Objekt um Ebenen zu definieren, die zusätzlich zur Editierebene nach dem Update- bzw. Insert-Vorgang neu geladen werden sollen (Komma separiert)

group = Gruppen Objekt, umschließt mehrere field Objekte und bildet einen Tab, Es muss mindestens ein group Element alle field Elemente umschließen!

title = XML-Attribut im group Objekt, Titel des Tabs

field = Feld-Objekt, beschreibt jeweils ein Datenbankfeld und kann beliebig oft definiert werden. Dieses Objekt umschließt die folgenden:

name = Spaltenname in der Postgistabelle

alias = Alias für die Bezeichnung der Spalte im Client

datatype = Datentyp der Spalte

gültige Typen:

int (Integer, ganze Zahl)

double (float, Werte mit Dezimalstellen)

text (string, Textwert)

boolean (true/false, vgl. boolvar)

edittyp = Typ des Feldes für Formular

gültige Typen:

edituser (zugewiesene Usergruppen des Objekts)

text (einzeiliges Textfeld, entspricht HTML-Element <input type="text">)

textarea (Mehrzeiliges Textfeld, entspricht HTML-Element <textarea>)

select (Auswahlliste; entspricht HTML-Element <select>)

checkbox (Checkbox; HTML-Element <input type="checkbox"> Dieses Element speichert den Default Wert des Feldes wenn aktiviert.

date (Datumsfeld mit Datumspicker, nur für datatype text).

noedit (kein Bearbeitungsfeld, lediglich nachrichtliche Anzeige, die Daten werden sind in einem Element über die Id "editform_feldname" im Dom abfragbar. Erscheint nur im Bearbeitungsmodus nicht bei Insert)

hidden (verstecktes Bearbeitungsfeld, die Daten werden in einem `` Element über die Id "editform_info_feldname" im Dom dargestellt. Erscheint bei insert und edit.)

exedit (verstecktes Bearbeitungsfeld, Möglichkeit der Anbindung externer Module über Javascriptfunktionen, vgl. `jsfunction` und `jsbuttonalias`, für das Feld wird ein verstecktes Feld erzeugt, welches mit aktualisiert wird. Erscheint nur im Bearbeitungsmodus nicht bei Insert)

jslink(kein Bearbeitungsfeld, bietet die Möglichkeit ein Javascriptfunktion wie im Query Template einzubinden, Funktioniert nur mit `linkprestring` und `linkpoststring`

html (kein Bearbeitungsfeld, bietet die Möglichkeit das Formular mit HTML zu gestalten. Achtung bei HTML Tags immer innerhalb des CDATA Elements! z.B. `<![CDATA[<hr>]]>`)

linkprestring = Zeichenkette die vor den `jslink` gesetzt werden kann

linkpoststring = Zeichenkette die nach den `jslink` gesetzt werden kann

jsfunction = Name der Javascript Funktion wenn Edittyp "exedit"

jsbuttonalias = Alias des Javascript Buttons wenn Edittyp "exedit"

htmleditor = (nur für edityp textarea) soll eine texarea als HTML Editor ausgegeben werden? mögliche Werte: true/false

selectvars = (nur für edityp select) Werte für Auswahllisten durch Komma separiert, keine Sonderzeichen oder Umlaute möglich!

selectalias = (nur für edityp select) Werte für Aliase zu `selectvars` durch Komma separiert! Muß zwingend der Reihenfolge und Anzahl der `selectvars` entsprechen! Angabe nur optional.

boolvar = Default-Wahr-Wert für ein Booleanfeld (Checkbox), unterstützte Werte: true, TRUE, t, 1, Standard:true

editwidth = Breite des jeweiligen Textfeldes oder Textarea (in Pixel!)

editrows = Höhe einer Textarea (entspricht HTML-Attribut cols)

editvalidate = soll das Feld auf einen Eintrag validiert werden, ist also Pflichtfeld? mögliche Werte: true/false

editdefaultvalue = Defaultwert der bei Eingabemaske eingetragen werden soll wenn Feld leer ist.
Möchte man den aktuell eingeloggten User als Defaultwert nutzen trägt man als Wert den Ausdruck 'edituser' ein.

Für jede definierte Datei muss ein edittyp „key“, ein edittyp „geom“ und ein edittyp „edituser“ definiert werden. Für den Key muss in PostgreSQL eine Sequenz für die automatische Erzeugung des Schlüssels vorhanden sein und eine Zahl sein. Die Sequenzen werden normalerweise beim Import eines Shapefile mit `shp2pgsql` erzeugt, können aber auch bequem im pgAdmin definiert werden. Ist das Feld edituser bei einer importierten oder vorhandenen Datei noch nicht vorhanden muss dieses angelegt werden und evtuell den Objekten über eine

Abfrage oder Verschneidung ect. der gewünschte edituser zugewiesen werden (vgl. dazu die Tabelle „admin“ in der Beschreibung der Clientdatenbank).

Im folgenden Beispiel wird ein Feld als Select-Feld mit drei vorgegebenen Einträgen und Validierung, ein Textfeld und ein Feld zur Bearbeitung von Integerwerten sowie die notwendigen Felder des Primärschlüssels der Geometriespalte und des Users definiert.

Beispiel eines Edit-Templates mit allen möglichen Datentypen:

```
<?xml version="1.0" encoding="utf-8"?>
<daten>
  <group title = "Tab 1">
    <field>
      <name>ttext</name> <!-- Edittyp text, Datentyp text -->
      <alias>Test text</alias>
      <datatype>text</datatype>
      <edittyp>text</edittyp>
      <editwidth>240</editwidth>
      <editvalidate>true</editvalidate>
    </field>
    <field>
      <name>tnoedit</name> <!-- Edittyp neoedit -->
      <alias>Test Noedit</alias>
      <edittyp>noedit</edittyp>
    </field>
    <field>
      <name>ttextarea</name> <!-- Edittyp textarea, Datentyp text -->
      <alias>Test Textarea HTML</alias>
      <datatype>text</datatype>
      <edittyp>textarea</edittyp>
      <htmleditor>true</htmleditor>
      <editwidth>240</editwidth>
      <editrows>3</editrows>
      <editvalidate>>false</editvalidate>
    </field>
    <field>
      <name>tdate</name> <!-- Edittyp date, Datentyp text -->
      <alias>Test Date</alias>
      <datatype>text</datatype>
      <edittyp>date</edittyp>
      <editwidth>120</editwidth>
      <editvalidate>true</editvalidate>
    </field>
    <field>
      <name>tint</name> <!-- Edittyp text, Datentyp int -->
      <alias>Test Integer</alias>
      <datatype>int</datatype>
      <edittyp>text</edittyp>
      <editwidth>50</editwidth>
      <editvalidate>true</editvalidate>
    </field>
    <field>
      <name>tdouble</name> <!-- Edittyp text, Datentyp double -->
      <alias>Test Double</alias>
      <datatype>double</datatype>
      <edittyp>text</edittyp>
      <editwidth>50</editwidth>
      <editvalidate>true</editvalidate>
    </field>
    <field>
      <name>tselect</name> <!-- Edittyp select, Datentyp text -->
      <alias>Test Select</alias>
      <datatype>text</datatype>
      <edittyp>select</edittyp>
      <selectvars>var1,var2,var3</selectvars>
```

```

        <selectalias>Test1,Test2,Test3</selectalias>
        <editvalidate>true</editvalidate>
    </field>
    <field>
        <name>tboolean</name> <!-- Edittyp checkbox, Datentyp boolean -->
        <alias>Test Boolean</alias>
        <datatype>boolean</datatype>
        <edittyp>checkbox</edittyp>
        <boolvar>true</boolvar>
    </field>
    <field> <!-- Edittyp html -->
        <edittyp>html</edittyp>
        <html><![CDATA[<hr>]]></html>
    </field>
    <field>
        <name>beschr</name>
        <datatype>text</datatype>
        <alias>Textarea</alias>
        <edittyp>textarea</edittyp>
        <editwidth>240</editwidth>
        <editrows>1</editrows>
    </field>
</group>
<group title = "Tab 2">
    <field>
        <name>label</name>
        <datatype>text</datatype>
        <alias>Text</alias>
        <edittyp>text</edittyp>
        <editwidth>240</editwidth>
    </field>
    <field>
        <name>lsize</name>
        <datatype>int</datatype>
        <alias>Textgröße</alias>
        <edittyp>text</edittyp>
        <editwidth>240</editwidth>
    </field>
    <field>
        <name>edituser</name>
        <alias>User</alias>
        <datatype>text</datatype>
        <edittyp>hidden</edittyp>
        <editdefaultvalue>edituser</editdefaultvalue>
    </field>
</group>
</daten>

```

2.7.3 Export-Modul Templates (mod_export)

template/export

export_list.xml

Diese Datei definiert bei aktivem Export-Modul die Parameter für den Vektor Export von Ebenen. Als Schnittstelle kommt die Software OGR2OGR zum Einsatz.

data = root-Objekt der XML-Datei (Wurzelement), umschließt alle anderen Objekte und darf nur einmal definiert werden

layer = Layer-Objekt, beschreibt jeweils einen Layer im Mapfile und kann beliebig oft definiert werden. Dieses Objekt umschließt die folgenden:

name = Name der Ebene (Pflichtelement)

alias = Alias für die Bezeichnung der Ebene (Pflichtelement)

dbcon = Name der Datenbankverbindungsdatei (ohne .php) im Ordner clientname/connections (Pflichtelement)

table = Name Datenbanktabelle / View (Pflichtelement)

schema = Name des Schemas in der die Datentabelle / View gespeichert ist. (Pflichtelement wenn abweichend von public)

id = Name der ID-Spalte (Pflichtelement)

fields = Falls nicht alle Spalten ausgegeben werden können über dieses Element ausgewählte Felder (Komma separiert) angegeben werden, dabei muss auch die Id-Spalte und die Geometriespalte mit angegeben werden! (optionales Element)

maxobj = Begrenzung der maximal zu exportierenden Objekte (Integer) (optionales Element)

groups = Angabe von ausgewählten Usergruppen (Komma separiert) denen der Export der Ebenen erlaubt wird, bei fehlendem Element darf jeder exportieren (optionales Element)

services = Angabe von ausgewählten Services (Komma separiert) in denen der Export der Ebenen möglich ist, bei fehlendem Element darf in jedem Service mit aktiviertem Export Modul exportiert werden. (optionales Element)

export_format.xml

Diese Datei definiert bei aktivem Export-Modul die Parameter für den Vektor Export von Ebenen. Als Schnittstelle kommt die Software OGR2OGR zum Einsatz.

data = root-Objekt der XML-Datei (Wurzelement), umschließt alle anderen Objekte und darf nur einmal definiert werden

format = Format-Objekt, beschreibt jeweils ein Format der OGR Schnittstelle. Dieses Objekt umschließt die folgenden:

name = Name des OGR-Formats (vgl. http://www.gdal.org/ogr_formats.html)

Case Sensitive! Welche Formate zum Export geeignet sind richtet sich nach der installierten GDAL Version. Abfrage über: [ogrinfo --formats]

alias = Alias für das Format in der Dropdownliste

extension = Erweiterung des Files im Dateisystem mit führendem Punkt

2.7.4 Metadaten-Modul Templates (mod_metadata)

template/metadata

metadata_list.xml

Diese Datei definiert bei aktivem Metadaten-Modul die Layer, welche Basiseinstellungen und Datenbankverbindungen konfiguriert sind
Folgende XML-Objekte sind dabei einstellbar:

data = root-Objekt der XML-Datei (Wurzelement), umschließt alle anderen Objekte und darf nur einmal definiert werden

client_title: Titel des Metadaten Clients

pgdbconnections = auswählbare Postgis Datenbankverbindungen. Die Verbindungsdaten müssen im Client-Verzeichnis "connections" als PHP file abgelegt sein. Dieses Objekt umschließt die folgenden:

db = Name bzw. Alias der Datenbank für das Dropdown

connectionfile = Name des PHP files im connection Ordner ohne .php

Beispiel eines Metadaten-Templates

```
<?xml version="1.0" encoding="UTF-8" ?>
<data>
  <client_title>Metadaten NETGIS</client_title>
  <pgdbconnections>
    <db>alkis</db>
    <connectionfile>db_connection_alkis</connectionfile>
  </pgdbconnections>
  <pgdbconnections>
    <db>netgis_client</db>
    <connectionfile>db_connection</connectionfile>
  </pgdbconnections>
</data>
```

2.7.5 WMS / WFS Dienste-Modul Templates (mod_ogc)

template/ogc

metadata_list.xml

Diese Datei definiert bei aktivem OGC-Modul die Basiseinstellungen und Basismetadaten der Dienstkonfiguration.

Folgende XML-Objekte sind dabei einstellbar:

data = root-Objekt der XML-Datei (Wurzelement), umschließt alle anderen Objekte und darf nur einmal definiert werden

client_title: Titel des OGC Clients

default_testbbox: Begrenzungskoordinaten der Bbox für den Test getMap Aufruf (Format [xmin, ymin, xmax, ymax] Komma separiert, möglichst quadratische Ausdehnung)

ogc_data= Basiskonfiguration, umschließt folgende Objekte:

ogc_configid = Id der Basis Konfiguration (z.B. "default" oder "config"), muss eindeutig sein möglichst ohne Leer- oder Sonderzeichen und klein geschrieben

max_extent maximaler Extent der WMS-Dienste Format [xmin ymin xmax ymax] mit Leerzeichen separiert

maxsize: Maximale Bildgröße die der WMS ausliefern darf (in Pixel)

proj_init: WMS-WFS Initial EPSG

ogc_srs: EPSG Liste der WMS- und WFS-Dienste mit Leerzeichen separiert

ogc_contactperson: Name der Kontaktperson

ogc_contactorganization: Name der Kontaktorganisation

ogc_contactposition: Position der Kontaktperson

ogc_contactvoicetelephone: Telefonnummer

ogc_contactfacsimiletelephone: Faxnummer

ogc_contactelectronicmailaddress: E-Mailadresse

ogc_adresstype: Adress Typ

ogc_address: Straße mit Hausnummer

ogc_postcode: Postleitzahl

ogc_city: Stadt

ogc_stateorprovince: Bundeslandkennung (z.B. Rlp oder BW)

ogc_country: Länderkennung (z.B.DE)

gml_geometries: Name der Geometrie, z.B. geometries (optional und experimentell für WFS 2.0)

wfs_maxfeatures_default: Maximale Anzahl die der WFS ausliefern darf als Default Wert

wfs_getfeature_formatlist: Optional weitere Formate die beim WFS mit angeboten werden: mögliche Formate: OGRGML,CSV,geojson,SHAPEZIP

Beispiel eines Metadaten-Templates

```
<?xml version="1.0" encoding="UTF-8" ?>
<data>
  <client_title>OGC-Geodatendienste NETGIS</client_title>
  <default_testbbox>316636,5496734,363062,5539950</default_testbbox>
  <ogc_data>
    <ogc_configid>default</ogc_configid>
    <max_extent>293310 5424000 464350 5644104</max_extent>
    <maxsize>3000</maxsize>
    <wfs_maxfeatures_default>50000</wfs_maxfeatures_default>
    <proj_init>25832</proj_init>
    <ogc_srs>EPSG:25832 EPSG:31466 EPSG:31467 EPSG:4326 EPSG:3857</ogc_srs>
    <ogc_contactperson>Sven Schröter</ogc_contactperson>
    <ogc_contactorganization>NETGIS GbR</ogc_contactorganization>
    <ogc_contactposition>Sachbearbeiter</ogc_contactposition>
    <ogc_contactvoicetelephone>0651-17047-31</ogc_contactvoicetelephone>
    <ogc_contactfacsimiletelephone>065117047-33</ogc_contactfacsimiletelephone>
    <ogc_contactelectronicmailaddress>info@netgis.de</ogc_contactelectronicmailaddress>
    <ogc_adresstype>postal</ogc_adresstype>
    <ogc_address>Benediktinerstr. 32a</ogc_address>
    <ogc_postcode>54292</ogc_postcode>
    <ogc_city>Trier</ogc_city>
    <ogc_stateorprovince>RP</ogc_stateorprovince>
    <ogc_country>DE</ogc_country>
    <gml_geometries>the_geom</gml_geometries>
    <wfs_maxfeatures_default>50000</wfs_maxfeatures_default>
    <wfs_getfeature_formatlist>OGRGML,CSV,geojson,SHAPEZIP</wfs_getfeature_formatlist>
  </ogc_data>
  <ogc_data>
    <ogc_configid>config1</ogc_configid>
    weitere Konfiguration...
  </ogc_data>
</data>
```

2.7.6 Plan-Modul Templates (mod_plan)

template/plan

plan_list.xml

Diese Datei definiert bei aktivem Plan-Modul die Layer, welche als Planverzeichnis dargestellt werden sollen.

Folgende XML-Objekte sind dabei einstellbar:

data = root-Objekt der XML-Datei (Wurzelement), umschließt alle anderen Objekte und darf nur einmal definiert werden

layer = Layer-Objekt, beschreibt jeweils einen Layer im Mapfile und kann beliebig oft definiert werden. Dieses Objekt umschließt die folgenden:

name = Name für den Layernamen

alias = Alias für den Layernamen

table = PostGIS-Tabellenname oder View in PostgreSQL

idfield = eindeutige Id der Datentabelle (in der Regel gid)

sortfield = name des Feldes nach dem im Suchergebnis sortiert werden soll

select = Ebene ist beim Start und nach Öffnen des Planmoduls eingestellt

checked = falls das Modul direkt im Service geladen wird und die Ebene bereits an sein soll hier true einstellen (optional)

symbol= hier kann ein symbol für den layer eingetragen werden (optional)

Diese Einträge (optional) müssen gleich in jedem layer stehen:

filterfield1 = Feld für zusätzliches Filter Dropdown

filterfield1_alias = Alias des Filter Dropdowns

filterfield1_vars = Variablen des Dropdowns Komma separiert

namefilter = true/false hiermit kann Suche nach dem Namen deaktiviert werden (Standard = true)

datefilter = true/false hiermit kann der Datumsfilter deaktiviert werden (Standard = true)

Beispiel eines Plan-Templates

```
<?xml version="1.0" encoding="UTF-8" ?>
<data>
  <layer>
    <name>plan_uebersicht</name>
    <alias>Bebauungspläne</alias>
    <table>bplan_poly</table>
    <idfield>gid</idfield>
    <sortfield>dat_ink</sortfield>
    <select>true</select>
    <filterfield1>stadt</filterfield1>
    <filterfield1_alias>Gemeinde</filterfield1_alias>
    <filterfield1_vars>Baldringen,Greimerath,Heddert,Hentern,Kell,Lampaden,Mandern,Paschel
,Schillingen,Schömerich,VG,Vierherrenborn,Waldweiler,Zerf</filterfield1_vars>
    <symbol>autoleg_plan_uebersicht.jpg</symbol>
    <namefilter>true</namefilter>
    <datefilter>true</datefilter>
  </layer>
  <layer>
    <name>fnp</name>
    <alias>Flächennutzungspläne</alias>
    <table>fnp_poly</table>
    <idfield>gid</idfield>
    <sortfield>dat_ink</sortfield>
    <filterfield1>stadt</filterfield1>
    <filterfield1_alias>Gemeinde</filterfield1_alias>
```



```

        <filterfield1_vars>Baldringen,Greimerath,Heddert,Hentern,Kell,Lampaden,Mandern,Paschel
,Schillingen,Schömerich,VG,Vierherrenborn,Waldweiler,Zerf</filterfield1_vars>
        <symbol>autoleg_fnp.jpg</symbol>
        <namefilter>true</namefilter>
        <datefilter>true</datefilter>
    </layer>
</data>

```

2.7.7 Print-Templates

template/print

in diesem Verzeichnis ist für jedes Drucktemplate eine XML-Datei hinterlegt. Dabei können beliebige Druckvorlagen definiert werden. Jedes Template bezieht sich dabei auf ein ISO-Format (DIN A4 - DIN A0 im Hoch oder Querformat).

format_list.xml

Diese Datei listet alle verfügbaren Drucktemplates und wird im Druckdialog als Dropdown-Liste geöffnet.

Folgende XML-Objekte sind dabei einstellbar:

daten = root-Objekt der XML-Datei (Wurzelement), umschließt alle anderen Objekte und darf nur einmal definiert werden

template = Template-Objekt, beschreibt jeweils ein Template und kann beliebig oft definiert werden. Dieses Objekt umschließt die folgenden:

alias = Name des Templates in der Dropdown-Auswahl des Druckdialogs

templatename = Name des Templates ohne Dateierweiterung .xml

services = optional, Namen der Services für die dieses Template ladbar ist Komma separiert. Ist das Objekt services nicht im Objekt template vorhanden ist das Template für alle Services ladbar.

templatename.xml

Die Templates werden jeweils in getrennten Dateien gespeichert.

Folgende XML-Objekte sind dabei einstellbar:

layout = root-Objekt der XML-Datei (Wurzelement), umschließt alle anderen Objekte und darf nur einmal definiert werden

map = definiert alle Basis-Objekte des Templates.

map->templateimg = Vorschaubild des Templates, sollte eine Breite von 200px nicht überschreiten und muss im Verzeichnis config/print liegen

map->format = DIN-Format (A4-A0)

map->orientation = Papier-Ausrichtung, erlaubte Parameter: P = Hochformat (Portrait), L = Querformat (Landscape)

map->width = Breite des Kartenfensters in mm

map->height = Höhe des Kartenfensters in mm

map->posy = Positionierung des Kartenfensters von oben in mm

map->posx = Positionierung des Kartenfensters von links in mm

map->scale = Maßstabsdefinition, erlaubte Parameter (als Liste durch Kommas separiert): currentextent = aktueller Bildausschnitt, currentscale = aktueller Maßstab, customscale = freier Maßstab, Zahl (integer) = feste Maßstabszahl beliebig oft

map->dpifactor = Faktor für die Auflösung ausgehend von 72 dpi (als Liste durch Komma separiert, darin wiederum der Faktor als Zahl (integer) und die Bezeichnung durch | separiert.

map->dynlegend->visible = mit diesem Objekt kann festgelegt werden, ob eine dynamisch angelegte Legende (vgl. legend_template.html) angezeigt werden soll (true) oder nicht (false). Zur Zeit wird diese Legende immer auf einer separaten Seite im DIN A4-Format an das Druckdokument angehängen.

map->dynlegend->checkbox = legt fest, ob im Druckdialog eine Checkbox erscheint (true/false), die es dem User ermöglicht zu wählen, ob eine dynamische Legende erzeugt wird oder nicht.

map->bordercoords->visible = legt fest, ob die Ausdehnungskoordinaten des Kartenausschnitts links unten und rechts oben mit ausgegeben werden (true/false).

map->refmap->visible = definiert, ob die Übersichtskarte mit ausgegeben werden soll (true/false).

map->refmap->width = definiert die Breite der Übersichtskarte in mm bei einer 0 wird die Breite automatisch proportional zur Höhe angepasst.

map->refmap->height = definiert die Höhe der Übersichtskarte in mm, bei einer 0 wird die Höhe automatisch proportional zur Breite angepasst.

map->refmap->posy = Positionierung der Übersichtskarte von oben in mm

map->refmap->posx = Positionierung der Übersichtskarte von links in mm

map->scalebar->visible = definiert, ob die Maßstabsleiste mit ausgegeben werden soll (true/false).

map->scalebar->width = definiert die Breite einer der vier Einheiten der Maßstabsleiste in mm

map->scalebar->height = definiert die Höhe der Maßstabsleiste in mm

map->scalebar->posy = Positionierung der Maßstabsleiste von oben in mm

map->scalebar->posx = Positionierung der Maßstabsleiste von links in mm

map->scalebar->textsize = die Textgröße der Beschriftung der Maßstabsleiste

map->scalebar->textspacing = Abstand der Beschriftung zur Maßstabsleiste

vartextcell = definiert alle variablen Text-Objekte des Templates. Kann beliebig oft definiert werden.

vartextcell->name = der Name der vor dem Eingabefeld im Druckdialog erscheint

vartextcell->postvar = Name der POST-Variablen über welche die Daten übersendet werden, darf keine Sonderzeichen, Quotes ect. enthalten. Dieser Name darf nur einmal pro Template vergeben werden.

vartextcell->editttype = gibt den Typ an, erlaubte Werte: text (einzeiliges Textfeld), textarea (Mehrzeiliges Textfeld), date (Automatisches Datum), select (Auswahlliste als Dropdown)

vartextcell->islegend = Soll der Text nur für die 2. Seite (Legende) gelten (true)? In diesem Fall wird die Standardüberschrift "Legende" nicht ausgegeben.

vartextcell->selectvars = Variablen Komma separiert falls editttype "select"

vartextcell->pretext = Text, der vor der Eingabe aus dem Textfeld erscheint

vartextcell->posx = Positionierung des Textes von links in mm

vartextcell->posy = Positionierung des Textes von oben in mm

vartextcell->textfont = Schriftart, erlaubte Schriften: helvetica, times, courier

vartextcell->textstyle = Schriftstil, erlaubte Werte: B = Fett; I = Kursiv, U = Unterstrichen, D =Durchgestrichen

vartextcell->textsize = Schriftgröße in Punkten (points)

vartextcell->textcolor = Schriftfarbe in RGB-Werten, Komma separiert

vartextcell->breaknum = Anzahl der Zeichen (incl. Leerzeichen) bis zum automatischen Zeilenumbruch

vartextcell->textspacing = Zeilenabstand bei mehreren Zeilen

textcell = definiert alle statischen Text-Objekte des Templates. Kann beliebig oft definiert werden. Die Unterobjekte entsprechen denen des Typs vartextcell.

img = definiert alle eingebundenen Bitmap-Dateien des Templates. Kann beliebig oft definiert werden. Zulässige Formate: GIF, JPEG, PNG, BMP bei aktivierter GD-Library.

img->file = Dateiname, alle eingebundenen Grafiken müssen im Ordner /config/print/ abgelegt sein!

img->width = definiert die Breite der Grafik in mm bei einer 0 wird die Breite automatisch proportional zur Höhe angepasst.

img->height = definiert die Höhe der Grafik in mm, bei einer 0 wird die Höhe automatisch proportional zur Breite angepasst.

img->posy = Positionierung der Grafik von oben in mm

img->posx = Positionierung der Grafik von links in mm

linebox = definiert alle Rahmen-Objekte des Templates. Kann beliebig oft definiert werden.

linebox->width = definiert die Breite des Rahmens in mm

linebox->height = definiert die Höhe des Rahmens in mm

linebox->posy = Positionierung des Rahmens von oben in mm

linebox->posx = Positionierung des Rahmens von links in mm

linebox->linewidth = Strichstärke des Rahmens in mm, Kommaangaben mit Punkt, z.B. 0.8

linebox->linedash = Zwischenabstand bei gestrichelten Linien (bei durchgezogenen Linien 0)

linebox ->linecolor = Linienfarbe in RGB-Werten, Komma separiert

linebox ->fillcolor = Füllfarbe des Rahmens in RGB-Werten, Komma separiert

2.7.8 Query-Templates



template/query



ebenennamen.xml

in diesem Verzeichnis können für jede abfragbare Ebene XML-Konfigurationsdateien abgelegt werden. Dabei muss der Dateiname dem Ebenennamen aus dem Map-File entsprechen.

Sie können so einfach definieren, welche Felder der Attributdaten abfragbar sind und diese mit einem entsprechenden Alias versehen. Soll das Feld als Link dienen muss der Wert in „link“ auf „true“ stehen und ein Alias für den Link vergeben werden.

Ist keine XML-Datei für die Ebene definiert, werden alle Felder der Attributtabelle ausgegeben.

Folgende XML-Objekte sind dabei einstellbar:

daten = root-Objekt der XML-Datei (Wurzelement), umschließt alle anderen Objekte und darf nur einmal definiert werden

noquerybuilder = optional, wenn der Querybuilder im Dialog der Ebene nicht erscheinen soll, erhält dieses Objekt den Wert „true“; soll der Querybuilder erscheinen, vergeben Sie den Wert „false“ oder entfernen das Objekt.

queryexcelexport = optional, per default hat dieses Objekt den Wert „true“ damit der Export-button erscheint, vergeben Sie den Wert „false“ zum entfernen des Buttons.

querypdfexport = optional, per default hat dieses Objekt den Wert „false“ damit der Export-button nicht erscheint, zum aktivieren vergeben Sie den Wert „true“.

showemptyfield = optional, per default hat dieses Objekt den Wert „true“, soll die Abfrage in der vertikalen Ausgabe die leeren Felder nicht anzeigen, vergeben Sie den Wert „false“.

field = Feld-Objekt, beschreibt jeweils ein Datenbankfeld und kann beliebig oft definiert werden. Dieses Objekt umschließt die folgenden:

name = Spaltenname des Tabelle (.dbf oder Postgistabelle)

alias = Alias für die Bezeichnung der Spalte in der Ausgabe des Abfrageergebnisses im Client

link = Wenn ein Link auf das Feld eingestellt werden soll erhält dieses Objekt den Wert „true“; ist kein Link eingestellt vergeben Sie den Wert „false“ oder entfernen das Objekt.

linkprestring = Zeichenkette (bzw. URL) die vor den Link gesetzt werden kann

linkpoststring = Zeichenkette die nach dem Link gesetzt werden kann (z.B. eine Dateiendung oder weitere URL-Variablen)

linkalias = Alias für die Bezeichnung des Links in der Ausgabe des Abfrageergebnisses im Client

linkjs = soll der Link als Javascript-Befehl geparkt werden, setzen Sie den Wert "true", ansonsten vergeben Sie den Wert „false“ oder entfernen das Objekt.

Der Befehl muss eventuell als CDATA eingesetzt werden, z.B.

```
<linkprestring><![CDATA[ javascript:openWindow('')]></linkprestring>
<linkpoststring>'</linkpoststring>
<linkalias>Details</linkalias>
<linkjs>true</linkjs>
```

linknotinclient= soll ein Link nicht in einer Clientversion ausgegeben werden?
Hilfreich um mögliche Konflikte bei angepassten Modullinks oder relativen URLs in der jeweiligen Variante zu umgehen (mögliche Werte: mobile,desktop)

button= Wird nur wirksam wenn link = true, styled den Link als Button

strreplace = Möglichkeit einen String zur Laufzeit zu ersetzen, zwei Zeichenketten durch Komma getrennt. Beispiel: xy,yz Hier wird im Abfragefeld jeder (Teil)String „XY“ durch „yz“ ersetzt.

roundnumber = soll ein numerisches Feld gerundet werden kann hier die Zahl der Stellen hinter dem Komma angegeben werden (Kaufmännische Rundung)

queryencoding = Encoding bei abweichendem Encoding zur globalen Variable \$queryencoding in baseconfig.php (z.B. <queryencoding>iso-8859-1</queryencoding>)

booleantrue= Alias im Falle eines boolean Feldes für TRUE (bei Postgresql als Standard t)

booleanfalse= Alias im Falle eines boolean Feldes für FALSE (bei Postgresql als Standard f)

styleclasstr= Name der CSS Style Klasse für das optionale Styling einer Attributzeile (HTML Tabellen tr des ausgegebenen Alias und Wert)

styleclasstd= Name der CSS Style Klasse für das optionale Styling einer Attributzeile (HTML Tabellen td des ausgegebenen Wertes)

Optional kann zusätzlich im daten-Objekt auch ein dbinfo Objekt konfiguriert werden (z.B. für das Editiermodul oder räumliche Querys).

dbinfo= dbinfo-Objekt, beschreibt jeweils die Eigenschaften der Datenbankverbindung und weiterer Einstellungen. Dieses Objekt umschließt die folgenden:

dbcon = Name der Datenbankverbindungsdatei im Verzeichnis Connections ohne Endung .php

table = Name der Quell-Datentabelle oder des Views

id = Name des Feldes mit dem Primärschlüssel der Tabelle

geom = Name der Geometriespalte der Tabelle

geomtyp = Geometrietyp der Tabelle, mögliche Einträge: point, line, polygon
Hinweis: Diese Einträge beziehen sich auch auf Multiobjekte.

spatialquery = Soll diese Ebene für räumliche Abfragen verwendet werden?
mögliche Einträge: true, false

spatialauth = Soll diese Ebene für im Falle einer Bearbeitung (mod_edit) einer räumlichen Authentifizierung unterliegen?
mögliche Einträge: true, false

spatialauth_dbcon = Name der Datenbankverbindungsdatei im Verzeichnis Connections ohne Endung .php worüber die räumliche Authentifizierung vorgenommen werden kann (hier liegt das Referenzpolygon)

spatialauth_table = Name der Tabelle oder Sicht (nur Polygonthemen) worüber die räumliche Authentifizierung vorgenommen werden kann

spatialauth_field = Name der Spalte in der Tabelle/Sicht in der die Gruppennamen für die räumliche Authentifizierung gespeichert sind

uploadtable = Name der Tabelle in welche die Upload Metadaten gespeichert werden. Die Tabelle muss folgende Struktur haben:

```
id serial NOT NULL,  
upload_id character varying,  
upload_file character varying,  
upload_info text,  
CONSTRAINT upload_lookup_pkey PRIMARY KEY (id)
```

uploadpath = absoluter Pfad zum Verzeichnis der hochzuladenden Dateien

uploadmaxpixel = Anzahl der Pixel auf die Bilder automatisch in ihrer größten Höhe bzw. Breite proportional skaliert werden soll

uploadmaxsize = Maximal zulässige Größe der Bilder in KB

uploadfiletyp = zulässige Filetypen z.Zt. nur img (= .png, .gif, .jpg, .jpeg)

edithidedelete= Sollen Objekte diese Ebene im Falle einer Bearbeitung (mod_edit) gelöscht werden können? mögliche Einträge: true, false

edithidenew= Sollen Objekte diese Ebene im Falle einer Bearbeitung (mod_edit) hinzugefügt werden können? mögliche Einträge: true, false

Das folgende Beispiel zeigt den Aufbau einer XML-Konfigurationsdatei für vier Felder der Attributtabelle. Für das zweite Feld wurde ein Link definiert. Zusätzlich wurde auch das Objekt dbinfo konfiguriert. Darin enthalten sind für das Editiermodul eine Konfiguration der Uploadfunktion. Das verwendete Feld upload_id zeigt zudem ein Beispiel wie Javascript-Funktionen als Link zum Einsatz kommen kann.

Beispiel eines Query-Templates:

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<daten>
  <dbinfo>
    <dbcon>db_connection_data</dbcon>
    <table>redline_polygon</table>
    <id>gid</id>
    <geom>geom</geom>
    <geomtyp>polygon</geomtyp>
    <spatialquery>true</spatialquery>
    <uploadtable>upload_lookup</uploadtable>
    <uploadpath>C:\inetpub\wwwroot\upload\bilder</uploadpath>
    <uploadurl>../upload/bilder</uploadurl>
    <uploadmaxpixel>800</uploadmaxpixel>
    <uploadmaxsize>3000</uploadmaxsize>
    <uploadfiletyp>img</uploadfiletyp>
  </dbinfo>
  <field>
    <name>feldname1</name>
    <alias>Meindatenalias1</alias>
  </field>
  <field>
    <name> feldname2</name>
    <alias> Meindatenalias2</alias>
    <link>true</link>
    <linkprestring>http://server.de/datei.php?var1=suchen&var2=</linkprestring>
    <linkpoststring>&var3=test</linkpoststring>
    <linkalias>Linkaliastext</linkalias>
  </field>
  <field>
    <name> feldname3</name>
    <alias> Meindatenalias3</alias>
  </field>
  <field>
    <name>upload_id</name>
    <alias>Bilder</alias>
    <link>true</link>
    <linkprestring><![CDATA[ javascript:openUploadPicture('mylayer',0,')]]></linkprestring>
    <linkpoststring>)</linkpoststring>
    <linkalias>Bilder öffnen</linkalias>
    <linkjs>true</linkjs>
  </field>
</daten>
```


3. Informationen zu verwendeter OpenSource Software

Mapserver

<http://mapserver.org>

Homepage zu UMN-Mapserver

<http://mapserver.org/de/mapscript/php/index.html>

Referenz zu Funktionen und Klassen von PHP-Mapscript

<http://www.mapserver.org/mapfile/>

Referenz zur Mapfile-Konfiguration

PostgreSQL / PostGIS

<http://www.postgresql.org/>

Homepage zur PostgreSQL- Datenbank mit Dokumentation und Download

<http://postgis.refractory.net/>

Homepage zur räumlichen PostgreSQL-Extension POSTGIS

<http://postgis.refractory.net/documentation/manual-svn/reference.html>

Postgis Referenz aller Funktionen

PHP

http://www.maptools.org/php_mapscript/index.phtml

PHP Mapscript zur Erweiterung von PHP für Mapserver-Funktionen

<http://www.tcpdf.org/>

PHP-Klasse zur dynamischen Erzeugung von PDF-Dokumenten

<http://www.verot.net/res/sources/class.upload.html>

PHP-Klasse für Upload-Funktionen

http://dl.maptools.org/dl/php_ogr/php_ogr_documentation.html

PHP-Extension zur Verwendung der OGR Simple Features Library

Javascript

<http://openlayers.org>

Javascript-Framework für den Mapping Bereich Openlayers Version 3

<http://openlayers.org/two/>

Javascript-Framework für den Mapping Bereich Openlayers Version 2

<https://jquery.com/>

mächtiges Javascript-Framework mit zahlreichen verwendeten Funktionen

<http://www.telerik.com/kendo-ui>

Javascript-Framework für das Client Frontend basierend auf jQuery