

# GeoShop Webserver Benutzerhandbuch

## Zusammenfassung

Diese Dokumentation beschreibt die Konfiguration des GeoShop Webserver.

Die Dokumentation darf nur mit Erlaubnis der infoGrips GmbH vervielfältigt werden.

# Inhaltsverzeichnis

1. Einleitung .....	4
1.1. Überblick .....	4
1.2. Aufbau dieser Dokumentation .....	4
1.3. Konventionen .....	4
2. Publizieren von Webseiten .....	5
2.1. Konfiguration in appserver.opt / AppServer .....	5
2.2. Konfiguration in appserver.opt / WebServer .....	5
3. Verschlüsselung mit HTTPS .....	7
3.1. Konfiguration in appserver.opt / AppServer .....	7
3.2. Serverzertifikate erstellen / installieren .....	8
3.2.1. Überblick .....	8
3.2.2. Selbst signiertes Serverzertifikat erstellen .....	8
3.2.3. Offizielles Serverzertifikat beantragen und installieren .....	9
3.3. Clientzertifikate .....	9
4. Redirector .....	10
4.1. Einleitung .....	10
4.2. Konfiguration in appserver.opt/Redirector .....	10
4.3. Redirector Anwendungsbeispiele .....	10
5. CGI-Schnittstelle .....	12
5.1. Einleitung .....	12
5.2. Installation .....	12
5.3. Ausführung von CGI-Skripts .....	12
5.3.1. OPT.temp_dir .....	12
5.3.2. Weitere Laufzeitoptionen .....	12
5.3.3. Debugging .....	13
5.3.4. MIME-Type .....	13
5.4. Beispiel .....	13

# 1. Einleitung

## 1.1. Überblick

Im **GeoShop Server** ist ein integrierter **Webserver** enthalten, welcher vom Benutzer für eigene Zwecke konfiguriert werden kann. Der Webserver wird aber auch vom GeoShop für die interne Kommunikation der diversen GeoShop-Module bzw. der GeoShop Server untereinander verwendet (z.B. ICS-Server oder Portalserver). Der GeoShop Webserver bietet jedoch weit mehr Funktionen als nur das Publizieren von Webseiten. Folgende Eigenschaften sind im GeoShop Webserver enthalten:

- **Publizieren von Webseiten** und grossen Dateien > 2 GByte.
- **HTTP-Authentifizierung** über GeoShop Benutzer.
- **Verschlüsselung** mit HTTPS.
- Integrierter **Redirector** mit dem Seiten aus anderen Webservern in den GeoShop Webserver eingeblendet werden können.
- **CGI-Schnittstelle** für iG/Script Programme.

Diese Dokumentation beschreibt die diversen Möglichkeiten der Webserver-Konfiguration.



Oft ist es nicht notwendig den integrierten Webserver speziell zu konfigurieren, da der Webserver für die wichtigsten Aufgaben des GeoShop bereits vorkonfiguriert ist (z.B. Publizierung von index.html oder Download von Bestellungen).

## 1.2. Aufbau dieser Dokumentation

Diese Dokumentation ist wie folgt aufgebaut:

- Kapitel 2: Beschreibt das Publizieren von Webseiten.
- Kapitel 3: Beschreibt die Verschlüsselung mit HTTPS.
- Kapitel 4: Beschreibt den Redirector.
- Kapitel 5: Beschreibt die CGI-Schnittstelle.

## 1.3. Konventionen

In dieser Dokumentation werden folgende Konventionen eingehalten:

Kursiv	Namen von Dateien und URL's
<b>fett</b>	neue Begriffe, Namen von Funktionen oder Methoden
courier	Programmtext oder Eingaben im Betriebssystem

## 2. Publizieren von Webseiten

### 2.1. Konfiguration in `appserver.opt` / `AppServer`

Im Abschnitt `AppServer` von `appserver.opt` kann man folgende Webserver Optionen setzen:

#### **HTTP\_PORT**

TCP/IP-Port unter welchem der GeoShop Webserver gestartet werden soll (Default = 3501). Falls die Option `HTTP_PORT` geändert wird, muss der GeoShop Server frisch gestartet werden.

#### **HTTP\_DEFAULT**

Defaultseite, welche bei nicht explizit im Webserver vorhandenen Webseiten ersatzweise angezeigt wird (z.B. `\www\index.html`).

#### **THREAD\_POOL\_SIZE**

Maximale Anzahl Threads (= Subprozesse), welche vom Webserver für die Verarbeitung von Anfragen verwendet wird (Default = 100). Dieser Wert muss nur bei extrem vielen Zugriffen auf den Webserver höher gesetzt werden.

#### **AUTHENTIFY**

Relative URL für welche HTTP-Authentifizierung verlangt wird (Default = `SWMS`). Die HTTP-Authentifizierung erfolgt über einen gültigen GeoShop Benutzer. Für `AUTHENTIFY` darf auch eine kommaseparierte Liste von URL's angegeben werden.

Beispiel:

```
AppServer MAP
...
HTTP_PORT STRING 80
HTTP_DEFAULT STRING \www\index.html
THREAD_POOL_SIZE STRING 200
AUTHENTIFY STRING test1,test2
}
```

### 2.2. Konfiguration in `appserver.opt` / `WebServer`

Vom GeoShop Webserver werden die Verzeichnisse `\user\www` und `\data\download` automatisch publiziert. Optional kann man mit dem Abschnitt `WebServer` in `appserver.opt` weitere Verzeichnisse im Webserver publizieren:

```
WebServer MAP
<local_url1> STRING <path1>
<local_url2> STRING <path2>
...
<local_urlN> STRING <pathN>
}
```

Die Parameter haben folgende Bedeutung:

#### **local\_url**

Lokale URL unter welcher das Verzeichnis `<path>` im GeoShop Webserver publiziert werden soll.

#### **path**

Verzeichnispfad relativ zu `GEOSHOP_DIR` oder absoluter Pfad.



Absolute Pfade sollten nur benutzt werden, wenn dies unbedingt notwendig ist. Ausserdem dürfen bei absoluten Pfaden keine Netzwerkpfade verwendet werden.

Beispiel:

```
WebServer MAP
  test STRING data\test
}
```

Publiziert alle Dateien im Verzeichnis GEOSHOP\_DIR\data\test unter der URL <http://localhost:3501/test>.



Der GeoShop Webserver sucht für path analog den INTERLIS-Tools zuerst im \user Ast und danach im \system Ast.

## 3. Verschlüsselung mit HTTPS

### 3.1. Konfiguration in appserver.opt / AppServer

HTTPS wird wie folgt im GeoShop aktiviert / gestartet:

1. `SSL_PORT` in `\user\options\appserver.opt` unter AppServer setzen:

```
AppServer MAP
...
SSL_PORT STRING 443
}
```

2. GeoShop neu starten. Am Schluss von `appserver.log` sollte die Meldung: `SSLServer started on port 443.` erscheinen.
3. Jetzt kann man bereits auf den GeoShop via HTTPS zugreifen (z.B. `https://localhost/client5/index.html` im Webbrowser aufrufen). Es ist auch möglich mit beliebigen GeoShop Client-Tools (z.B. Administrator oder Ordermanager) auf den GeoShop zu zugreifen. Dazu muss man beim Login im Feld Server die URL `https://localhost` eingeben.



Vor der GeoShop Version 2022.0 wurde der Zertifikatspeicher im Format JKS (Java Key Store) im Verzeichnis `GEOSHOP_USER_DIR\ssl` abgespeichert. JKS ist ein proprietäres Format und wurde daher in der GeoShop Version 2022.0 durch das standardisierte Format PKCS12 (.pfx) ersetzt.



SSL-Zertifikate einer offiziellen CA werden normalerweise im PKCS12 Format (.pfx) geliefert.



Alte Zertifikatspeicher im JKS Format werden beim GeoShop Start automatisch nach PKCS12 übersetzt. Passwörter und Zertifikat-Alias werden dabei auf Default Werte gesetzt.

Vollständige Liste aller im Abschnitt AppServer unterstützen SSL-Optionen:

#### **SSL\_PORT (r)**

Port auf dem der SSL-Server gestartet werden soll, normalerweise 443 (Standard Port von HTTPS). Die Angabe dieser Option ist obligatorisch, sonst wird der SSL-Server vom GeoShop nicht gestartet.

#### **SSL\_PROTOCOL**

Verwendetes SSL-Protokoll (Default: TLSv1.2).

#### **SSL\_STORE (o)**

Name des .pfx Zertifikatspeicher in dem das Zertifikat bzw. der Private Key gespeichert ist (Default: `ssl.pfx`).

#### **SSL\_STORE\_PASSWORD (o)**

Password für den Zertifikatspeicher.

#### **SSL\_ALIAS (o)**

Alias unter dem das Zertifikat im SSL-Zertifikatspeicher abgelegt ist (Default: `geoshop`). Muss nur angegeben werden, wenn mehr als ein Zertifikat im SSL-Zertifikatspeicher abgelegt ist. Wenn das angegebene Zertifikat im Zertifikatspeicher nicht existiert, wählt der GeoShop das erste Zertifikat im Zertifikatspeicher.

#### **SSL\_KEY\_PASSWORD (o)**

Password für den Private Key im Zertifikatspeicher (Default: `SSL_STORE_PASSWORD`).

## 3.2. Serverzertifikate erstellen / installieren

### 3.2.1. Überblick

Der GeoShop verwendet standardmässig ein selbst signiertes und für die Domäne localhost ausgestelltes Serverzertifikat für die Verschlüsselung der Daten. Der Zertifikatspeicher ist unter `\user\certs\ssl.pfx` abgelegt. Wenn man ein Serverzertifikat installieren will, hat man folgende Möglichkeiten:

1. Man generiert ein selbst signiertes Serverzertifikat für seine Domäne. Diese Methode hat den Vorteil, dass sie schnell durchgeführt werden kann und keine wiederkehrenden Kosten nach sich zieht. Die Benutzer erhalten jedoch im Browser eine Fehlermeldung, welche vom Benutzer übersprungen werden muss (s.a. Abschnitt Clientzertifikat). Diese Methode eignet sich daher vor allem für Testumgebungen und nicht produktive Webseiten. Das selbst erstellte Zertifikat ist (praktisch) unbeschränkt gültig.
2. Man bestellt ein Zertifikat für die gewünschte Domäne bei einer offiziellen Zertifizierungsstelle (CA). Darauf installiert man das offizielle Serverzertifikat im GeoShop. Sofern das Zertifikat von einer geläufigen CA (z.B. Verisign, Thawte, TC Trustcenter, etc.) erstellt wurde, erhalten Benutzer im Browser keine Fehlermeldung mehr. Allerdings muss man für die Erstellung und Verwendung des Serverzertifikat eine wiederkehrende Gebühr entrichten.



Der Name der Domäne wird im GeoShop unter `AppServer.HOST` angegeben (z.B. `www.infogrips.ch`).



Von einer CA signierte Serverzertifikate sind nur noch ein Jahr gültig und müssen rechtzeitig erneuert werden um Fehlermeldungen im Browser der Benutzer zu vermeiden.

Weil es für beide Methoden Anwendungsfälle gibt, sind im nachfolgenden beide Methoden beschrieben.

### 3.2.2. Selbst signiertes Serverzertifikat erstellen

Erstellen oder ergänzen Sie den Abschnitt `MetaInfo` in `appserver.opt` wie folgt:


```
MetaInfo MAP
  Provider STRING <organization>
  Unit STRING <organizational-unit>
  Street STRING <street>
  ZIP STRING <zip>
  City STRING <city>
  State STRING <state>
  Country STRING <country>
}
```

Nachfolgend ein vollständig ausgefülltes Beispiel:

```
MetaInfo MAP
  Provider STRING 'infogrips GmbH'
  Unit STRING 'Development'
  Street STRING 'Technoparkstrasse 1'
  ZIP STRING '8005'
  City STRING 'Zuerich'
  State STRING 'ZH'
```



```
Country STRING 'CH'
}
```

 Falls man bereits für eine ältere Version des GeoShop einen Zertifikatspeicher im JKS Format erstellt hat, muss man nicht unbedingt den Abschnitt `MetaInfo` ausfüllen, da diese Angaben aus dem JKS Zertifikatspeicher entnommen werden. Weil `MetaInfo` jedoch auch für andere GeoShop Funktionen verwendet wird (z.B. Portalserver), empfehlen wir trotzdem die Erstellung von `MetaInfo`.

Nun kann ein selbst signiertes Serverzertifikat wie folgt erstellt werden:


1. Löschen Sie die Datei `\user\certs\ssl.pfx`, falls diese existiert.
2. Starten Sie den GeoShop neu.
3. Der GeoShop erzeugt nun für die Domäne `AppServer.HOST` aus den Angaben im Abschnitt `MetaInfo` automatisch den Zertifikatspeicher `\user\certs\ssl.pfx` und lädt das darin enthaltene Serverzertifikat.


### 3.2.3. Offizielles Serverzertifikat beantragen und installieren

Das Serverzertifikat muss zuerst bei einer offiziellen CA im PKCS12 Format bestellt werden. Die CA liefert dann das von ihr signierte Serverzertifikat als `.pfx` Datei (manchmal ist die Dateiendung auch `.p12`), welche man wie folgt im GeoShop installiert:

1. Kopieren Sie die `.pfx` Datei nach `\user\certs\ssl.pfx`.
2. Passen Sie die notwendigen `SSL_` Parameter im Abschnitt `AppServer` an, z.B.

```
AppServer MAP
...
SSL_PORT STRING 443
SSL_STORE_PASSWORD STRING <store-password>
SSL_KEY_PASSWORD STRING <key-password>
```

 Die Passwörter `<store-password>` bzw. `<key-password>` werden von der CA geliefert.

 `<key-password>` muss nur angegeben werden, wenn der Private Key von der CA durch ein zusätzliches Passwort gesichert wurde (ist selten der Fall).

3. Starten Sie den GeoShop neu.

## 3.3. Clientzertifikate

Dieser Abschnitt gilt nur für "selbst signierte Serverzertifikate" (s.a. oben).

Für die Administratorwerkzeuge müssen keine Clientzertifikate installiert werden.

Falls man mit einem aktuellen Internet Browser (Edge, Chrome, Firefox oder Safari) via HTTPS auf den GeoShop zugreifen will, gibt es eine Fehlermeldung betreffend fehlender Sicherheit des Serverzertifikats, welche man überspringen muss.

## 4. Redirector

### 4.1. Einleitung

Mit dem Redirector können Webseiten von externen Webservern im GeoShop Webserver eingebunden werden. Die externen Webseiten erscheinen für den Benutzer des GeoShop Webserver wie Seiten, welche *direkt* vom GeoShop Webserver geliefert werden. Der GeoShop Webserver agiert damit quasi als Proxyserver zwischen dem Client und der externen Webseite.

### 4.2. Konfiguration in appserver.opt/Redirector

Der Redirector kann im Abschnitt Redirector von appserver.opt wie folgt konfiguriert werden:

```
Redirector MAP
  <local_url1> STRING <external_url1>[,<user1>,<password1>]
  <local_url2> STRING <external_url2>[,<user2>,<password2>]
  ...
  <local_urlN> STRING <external_urlN>[,<userN>,<passwordN>]
}
```

Die Parameter haben folgende Bedeutung:

**local\_url**

Lokale URL im GeoShop Webserver.

**external\_url**

URL des externen Webserver inkl. http:// bzw. https://.

**user**

Benutzer für Basic-Authentification (optional).

**password**

Passwort für Basic-Authentification (optional).

Falls man nicht sicher ist, ob der Redirector korrekt konfiguriert ist kann man mit AppServer.RE-DIRECTOR\_DEBUG STRING ON (Default: OFF) Meldungen zur Weiterleitung in appserver.log aktivieren.

Beispiel:

```
Redirector MAP
  meta STRING https://www.infogrips.ch/geoshop/meta
}
```

Mit der obigen Definition kann man z.B. via `http://localhost:3501/meta/client5/index.html` auf den GeoShop Client des infoGrips Metashop zugreifen.

### 4.3. Redirector Anwendungsbeispiele

Zum Schluss sind noch ein paar Anwendungsbeispiele zum GeoShop Redirector zusammen gestellt:

**Publizieren von mehreren unabhängigen GeoShop Installationen auf Port 80**

Manchmal möchte man auf dem gleichen Server mehrere GeoShop Server installieren, aber trotzdem alle diese Server via Port 80 erreichen. Das ist normalerweise nicht

möglich, weil jeder GeoShop Server einen eigenen IP-Port für den HTTP-Server benötigt. Man kann sich nun wie folgt behelfen:

- Ein GeoShop Server wird auf Port 80 eingerichtet (= Proxyserver), die anderen GeoShop Server auf beliebigen Ports.
- Im Proxyserver richtet man den Redirector so ein, dass die anderen GeoShop Server als Subadressen im Proxyserver erscheinen.

### **Zugriff auf HTTPS verschlüsselte Webseiten in ICS-Skripts**

Das infoGrips Conversion System (ICS) hat eine eingebaute `SOCKET` Klasse mit welcher man auf externe Webseiten zugreifen kann. Die `SOCKET` Klasse unterstützt aber im Moment kein HTTPS. Dazu richtet man im GeoShop einen Redirect auf die externe mit HTTPS verschlüsselte Webseite ein. Der ICS-Skript kann dann via den GeoShop per HTTP auf die externe Webseite zugreifen.

### **Zugriff auf geschützte externe WMS-Seiten**

Im GeoShop WMS-Server ist auch ein WMS-Client eingebaut. Der WMS-Client kann externe WMS-Seiten im GeoShop WMS-Server verfügbar machen. Der WMS-Client unterstützt jedoch keine HTTP-Authentifizierung. Mit den Redirector Optionen `<user>` und `<passwort>` kann man dieses Problem elegant lösen, indem man die WMS-Anfragen via den Redirector an den externen WMS-Server weiter leitet.



Es ist grundsätzlich möglich mehrere GeoShop Redirector miteinander zu verbinden. Der kreativen Anwendung des Redirector werden hier also keine unnötigen Hindernisse in den Weg gelegt.

## 5. CGI-Schnittstelle

### 5.1. Einleitung

Mit der CGI-Schnittstelle ist es möglich iG/Script Programme im Webserver auszuführen um z.B. dynamische Webseiten zu generieren (z.B. als Resultat einer DB-Anfrage). Damit ein CGI-Script vom Webserver ausgeführt werden kann, muss es die Dateiendung `.igs` aufweisen (z.B. `test.igs`). CGI-Skripts werden normalerweise im Verzeichnis `\user\www\igs` abgelegt. Es kann jedoch auch jedes andere vom Webserver publizierte Verzeichnis für die Ablage von CGI-Skripts verwendet werden.

### 5.2. Installation

Damit CGI-Skripts im GeoShop ausgeführt werden können, muss zuerst im GeoShop Server der Dienst `igs` einem oder mehreren ICS-Servern zugewiesen und gestartet werden (s.a. GeoShop Administrator bzw. Konfigurationsdatei `\user\services\services.srv`).

### 5.3. Ausführung von CGI-Skripts

Wenn ein Benutzer auf die URL eines CGI-Skripts zugreift, wird der Skript vom GeoShop Webserver mit Hilfe von `ics.exe` ausgeführt. Der Standardoutput des CGI-Skripts wird dann als Resultat vom GeoShop Webserver zurück gegeben.

#### 5.3.1. `OPT.temp_dir`

Für temporäre Dateien erzeugt der Webserver vor dem Aufruf des Skripts automatisch ein temporäres Verzeichnis. Der Pfad zu diesem Verzeichnis wird dem Skript als `OPT.temp_dir` übergeben. Es ist wichtig, dass ein CGI-Skript temporäre Zwischenresultate in das Verzeichnis `OPT.temp_dir` schreibt. Sonst kann es zu Konflikten mit anderen gleichzeitig laufenden CGI-Skripts kommen. Nach der Ausführung des CGI-Skripts wird das Verzeichnis `OPT.temp_dir` automatisch wieder gelöscht.

#### 5.3.2. Weitere Laufzeioptionen

Neben `OPT.temp_dir` stellt der GeoShop dem CGI-Script folgende zusätzliche Laufzeioptionen zur Verfügung:

##### **OPT.URL**

Relative URL des aufgerufenen `.igs` Skript.

##### **Client HTTP-Parameter**

Alle HTTP-Parameter welcher der aufrufende HTTP-Client schickt in der Form `OPT.<HTTP-Param>` (z.B. `OPT.Host`, `OPT.User-Agent`, `OPT.Accept-Encoding`, etc.).

##### **ICS Optionen**

Alle Optionen, welche ein ICS-Skript zur Laufzeit erhält (z.B. `OPT.user_dir`, `OPT.system_dir`, `OPT.data_dir`, etc.).



Auf die Optionen des GeoShop Server in `appserver.opt` kann über die Map `OPTIONS` zugegriffen werden. Dazu muss aber vorgängig die Bibliothek `\script\util.lib` im Skript eingebunden werden.

### 5.3.3. Debugging

Für das CGI-Debugging steht folgende Option im Abschnitt AppServer von `appserver.opt` zur Verfügung:

#### IGS\_DEBUG

Debugging von CGI-Skripts ein- (ON) oder ausschalten (OFF) (Default = OFF). Wenn die Option auf ON gesetzt wird, wird der Debugging Output in die Datei `\data\temp\igs.log` geschrieben (s.a. CGI-Schnittstelle).

### 5.3.4. MIME-Type

Der Standard MIME-Type eines CGI-Skripts ist `text/xml`, da die meisten CGI-Skripts im GeoShop für die Generierung von XML-Dateien verwendet werden (z.B. SOAP-Schnittstelle oder `getCapabilities.igs` des WMS-Server). Falls ein CGI-Skript einen anderen MIME-Type zurück geben soll (z.B. `text/html`), muss man als letzten Befehl im Skript folgende Zeile einfügen:

```
OPT.mime_type '<MIME-Type>' SERIAL.SAVE_OBJECT
```

## 5.4. Beispiel

Zum Abschluss ist hier noch ein vollständiges Beispiel für einen CGI-Skript angegeben:

```
|LICENSE \license\geoshop.lic

DISPLAY '<html>'
DISPLAY '<head>'
DISPLAY '<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">'
DISPLAY '</head>'
DISPLAY '<body>'
DISPLAY '<title>hello, world!</title>'
DISPLAY '</body>'
DISPLAY '</html>'

OPT.mime_type 'text/html' SERIAL.SAVE_OBJECT
```

Falls der obige Skript unter `\user\www\igs\hello.igs` abgespeichert wird, erzeugt der Aufruf der URL `http://localhost:3501/igs/hello.igs` im Webbrowser den folgenden Output:

```
hello, world!
```