Universität Rostock Traditio et Innovatio

Professur für Geodäsie und Geoinformatik
Prof. Dr.-Ing. Ralf Bill
M.Sc. Markus Berger

Tutorial: OGC Web Services

Standardisation / Interoperability

This course unit describes common OGC web services and explains their use in the processing of open geodata.

Tips and tricks for using OGC-WxS will also be collected.

Introduction

Anyone who has ever worked with geodata knows the problem: If you want to carry out analyses on current issues, it is difficult to find data sets that are on the same time level. It is in the nature of geodata that they have a relation to reality. And since reality is subject to constant change, all data sets that do not explicitly describe a historical state must be updated frequently. If one works "classically", i.e. with local files and folders, this process is time-consuming. In practice, different versions of the same data set are often used in different places, which can quickly lead to discrepancies in analysis results.

How to solve this problem? If you look at the world outside of geodata, the best tool to get up-to-date information is the World Wide Web. This applies not only to news and trends, but also to data collections. And of course we also want to use these possibilities for the analysis of geodata. More specifically, we are concerned with the use of web services.

The entire structure of the Internet is based on services, which are usually implemented via a client-server relationship. So the basic structure exists, only suitable interfaces have to be designed for geodata.

Of course, everyone could now follow their own implementation. But just as we work with a collection of standardized file formats, we also want to standardize our services. Well known is the International Standard Organization (ISO), which takes care of industry standards worldwide. The Open Geospatial Consortium (OGC) was founded in 1994 especially for spatial information processing. OGC works closely with ISO to standardize data formats and processes in the field of geospatial data worldwide. This collaboration has resulted in standards and specifications for the web services we need. This is typically referred to as "OGC Web Services".

In this document we want to take a closer look at some of these services. First, however, a list of the currently defined interfaces:

Service name	*Abbreviation*	*Usage*
Web Map (Tile) Service	WM(T)S	Delivers web maps as image files, either in a selected image section, or in predefined tiles
Web Feature Service	WFS	Provides spatial vector data
Web Processing Service	WPS	Performs spatial operations on requests and returns the result
Web Coverage Service	WCS	Delivers spatial raster data in different formats
Web Catalogue Service	WCAS	Access and search for metadata
Web Gazetteer Service	WGS	Search for geodata using names and words
Web Coordinate Transformation Service	WCTS	Provides coordinate transformations
Web Pricing and Ordering Service	WPOS	Provide price and license information for geodata, and allow direct ordering
Web View Service	WVS	Provides 3D scenes
Web Terrain Service	WTS	Provides 3D terrain

For most of these services, not only standards but also technical specifications and reference implementations are available. In particular, the first entries in this table have long been used intensively in geo data processing.

Web Request Basics

All web services considered here are based on the Hypertext Transport Protocol (HTTP). This is one of the basic building blocks of the web, and allows different types of requests to be made to servers. Most of these methods retrieve data from a server, transfer new data, or modify what is already present on the server. In our case, we want to send certain data to the server so that it can give us the appropriate response. Mostly these data are collections of parameters, e.g. the coordinates of a rectangle defining a map section.

Tehre are two request methods to transfer such data: GET and POST. With GET, the parameters are written directly to the URL. At POST they are sent in the "body" of the http request. The URL is not modified, instead this type of request behaves more like a classic file upload/download and is especially suitable for larger requests. In both cases, the parameters are described by key-value pairs. Keys specify the name of the parameters (not case-sensitive), values the values of these parameters (case-sensitive). As an example we would like to send a request to the website http://server.com/page.html.

- GET: We write "http://server.com/page.html?param1=val1&param2=val2" in the address line of our browser and open the page.
- POST: On a website, two form fields called "Param1" and "Param2" are offered, as well as a button to submit their contents. The source code of the page is used to send a message to "http://server.com/page.html" after the button is clicked.

Selected OGC Web Services

We now want to take a closer look at the most frequently used OGC Web Services in practice.

WMS

The Web Map Service returns map sections upon request as pre-fabricated image files. In view of our knowledge about HTTP requests, the question now arises: What parameters does the server need?

First there is the so-called "Service" parameter. It determines which service is requested if a server offers other service interfaces in addition to the WMS, for example. Furthermore "Request" is needed. It determines exactly which function of the WMS is required. In addition to the direct data request, Web services can also list, for example, what functions they have exactly. In addition, "Version" can be used to specify for which WMS version a request is intended. This prevents that older automated processes are suddenly no longer functioning after system updates.

In order to describe the functionality of a WMS, the various possible "requests" are of particular interest, so we will continue with these:

GetCapabilities

This request returns an ISO-standardized XML metadata document. It contains information about what image formats the server can return, in which coordinate systems data is available, which data layers can be queried, and so on.

GetMap

The actual core of a WMS. This request returns a raster image of a specified area. This area and the appearance of the raster image are defined using various parameters. A selection of keys:

- BBOX - the desired map section as a rectangle.
- CRS - coordinate reference system, mostly as EPSG code.
- WIDTH/HEIGHT - the desired size of the image on the screen
- LAYERS - which card layers should be printed on the image
- STYLES/SLD - the desired style of the card
- …

A GET-based GetMap request, e.g. to get the districts of Rostock from the geodata offer of the Hanseatic city of Rostock, could look like this:

https://geo.sv.rostock.de/geodienste/ortsteile/wms?SERVICE=WMS&VERSION=1.3.0&REQUEST=GetMap&LAYERS=hro.ortsteile.ortsteile&WIDTH=512&HEIGHT=512&CRS=EPSG:3857&BBOX=1334743.3879,7180341.5631,1364362.7364,7209540.5079&FORMAT=image/png

At first glance, such a request can be quite overwhelming. But let's go through the individual parameters step by step: We put an inquiry to the side https://geo.sv.rostock.de/geodienste/ortsteile/wms. So this is a WMS, which offers districts of the Hanseatic city of Rostock. Nevertheless probably on this URL the WMS is the only service, we must specify over "SERVICE" that we want to use WMS functionalities. We need these in the "VERSION" 1.3.0 of the OGC WMS specification. Since we want to get a map image our "REQUEST" is GetMap. In the case of "LAYERS" the required value is more difficult to determine, here we have to look into the exact layer definition via GetCapabilities. The local part layer is called hro.ortsteile.ortsteile on this WMS. "WIDTH" and "HEIGHT" are set to 512, so we will get a square image with corresponding pixel resolution. The used "CRS" is EPSG:3857, a projected coordinate system, which is used in many internet

applications (called "Web-Mercator"). The "BBOX" defines two coordinates in this system, which stand for left-bottom and right-up in the desired rectangle of the terrain section. Finally it is stated that we want to get the result in the "FORMAT" PNG (Portable Network Graphics).

If you copy the above GetMap request into a browser and open the page, you get the following PNG file as result:

With "STYLE" and other parameters you can further manipulate the result. WMS uses so-called SLD files (Styled Layer Descriptor) for styles. Describing their syntax, however, would lead too far at this point. A comprehensive tutorial can be found at

http://docs.geoserver.org/stable/en/user/styling/sld/cookbook/.

DescribeLayer

(Optional) Returns detailed information about individual layers, if available.

GetFeatureInfo

(Optional) Returns the data basis for a selected area. Simplified variant of a WFS request.

GetLegendGraphics

(Optional) Similar parameters as GetMap, but instead of the map section it returns a legend for the selected image area. This is also an image file.

Note: WMS is very easy to use when creating web pages. No complex WebGIS has to be set up to display the maps correctly. Instead, the output image files can simply be superimposed using HTML and CSS to create complex maps with multiple layers and legends.

WFS

If we need not only the finished map representation, but also the underlying data in a suitable geodata format, then a WFS must be used instead of a WMS. Here no image file is returned, but vector data. The data is normally returned in the Geography Markup Language (GML), a format designed by the OGC and standardized by ISO for the storage of geodata in XML data structures. However, other formats such as GeoJSON can also be defined.

A request is similar to a WMS, but there are some new parameters and requests:

GetCapabilities

Same functionality as in WMS.

DescribeFeatureType

Returns standardized descriptions of the various object types offered by the service and their structure.

GetFeature

The actual data call, analogous to GetMap. WFS and WMS share basic parameters like "BBOX". Instead of "LAYERS" you use "TYPENAMES" to define which types of vector data you need. Analogous to the WMS you can filter these spatially via "BBOX". Further offered filter parameters are e.g. "COUNT" and "SORTBY", but there is also a "FILTER" parameter with which more complex sequences can be defined in GML. Their functionality is similar to that of SQL queries. If we only want to get streets of type "1" from a WFS, the filter could look like this:

```
<Filter>
    <PropertyIsEqualTo>
        <PropertyName>roadtype</PropertyName>
        <Literal>1</Literal>
    </PropertyIsEqualTo>
</Filter>
```

A POST request must be used to be able to write down a filter textually in this form. If you want to use this filter via GET, the corresponding special character codes must be used in the URL:

%3C Filter**%3E%3C**PropertyIsEqualTo**%3E%3C**PropertyName**%3E**roadtype**%3C**/PropertyName**%3E%3C**Literal**%3E**1**%3C**/Literal**%3E%3C**/PropertyIsEqualTo**%3E%3C**/Filter**%3E**

Frequent data processing operations can thus be applied even before the data set reaches the user.

WPS

However, the filter operations offered in WFS are only the beginning. In the age of Cloud Computing, it should quickly become clear that much more advanced operations can be made possible through requests to servers. The Web Processing Service was specified for this purpose. With it, it is possible to set up services that start complex processing on the server on request and then send the result back to the requestor.

The operations themselves are not standardized and can be implemented and offered on the server as required. Instead, as with the other services, only the interfaces for entering and outputting parameters and data are standardized.

GetCapabilities

Same functionality as in the other services. Lists, among other things, the possible operations (processes).

DescribeProcess

Once you have discovered the desired process in the capabilities, you write its ID in this query. The answer then contains specific information about how the process works and what input and output options are offered. Input and output are captured in GML structures just like in WFS, unless the output is a single file, such as an image file. This can be sent directly as a raw file.

Execute

The start of the actual process. These can run asynchronously and over long periods of time. First the user has to define what his inputs are, then how the answer should look like (in GML or raw data) and if the answer should be reachable via a web address afterwards. In addition, there are other connection points to obtain web addresses that provide information about the status of a long-running operation. This part of the WPS syntax is one of the more complex parts of the specification, and would go beyond the scope of this document. Most of the tools that offer to work with WPS bring with them simplified "Request Builders" that make working with Web operations possible even for less technically savvy users.

The inclined reader may notice at this point that, with a few exceptions, all W*S queries and answers run through the standardized GML. This standardization, which concentrates on XML documents, enables a very interesting procedure called "service chaining". Because of the common interfaces, chains of WPS operations can be created that may even use the output of other WFS or WMS as input. If the OGC web services are used to their full potential, entire processing chains can be defined over the Internet, which ensure that each step of the process always works with the most up-to-date data. Behind the display of a simple output card on a website, there could be dozens of complex WebGIS analysis steps without the user ever noticing.

WCS

As mentioned before, a WFS only offers vector data, a WMS only offers simple image files. But what if we want to work with georeferenced raster data sets? For this there is the "Web Coverage Service" (WCS). The possible requests are analogous to WFS: GetCapabilities, DescribeCoverage, GetCoverage. Just as with WFS, there is also the possibility to have filter operations performed by the server. So you can specify time periods, spectral bands and height restrictions before the result arrives. The possible output formats of a WCS include simple image files like the WMS, but also georeferenced formats like GeoTIFF or ArcGrid.

WCAS

The "Web Catalogue Service" (WCAS), partly also called "Catalogue Service for the Web" (CSW), allows the provision and search of metadata about geodata sets and spatial processes. In large spatial data infrastructures that are compatible with OGC web services, WCAS can be a central point of contact. These metadata can contain a variety of information. A selection: Identification numbers, restrictions, quality, timeliness, spatial references, validity, extension, etc.

WCAS benefit from the document-based architecture that GML brings with it. As an offshoot of XML, search and filter operations can be defined with a specially defined query language called XPath, which allows powerful search operations even on large XML datasets.

The WCAS standard is intended for operators, data suppliers as well as users and contains besides search and filter operations also possibilities to update metadata, or even to take over metadata from other WCAS standard compliant WCAS. This allows networks of information to be built in which records from a variety of sources can be merged without confusion about who originally owns a record. Especially in times when a mixture of open and commercial data is used for many GIS projects, such source information is important.

Use of OGC services in GIS

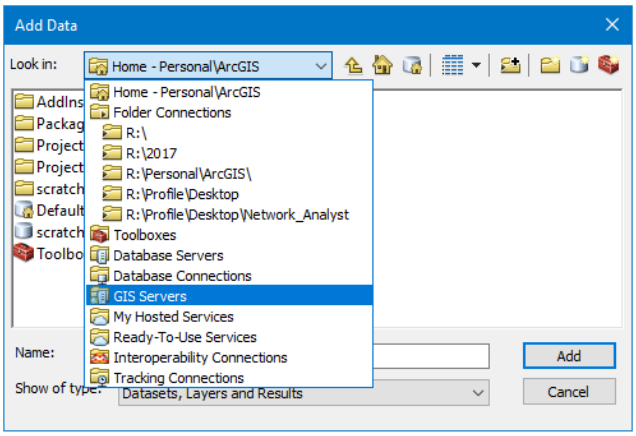
The mixture of HTTP requests and GML documents presented in the last section is obviously not a very user-friendly interface. Although they are readable enough to be used directly by humans, a lot of background knowledge is required.

Most GIS, whether on the web or with a desktop client, support easy working with OGC Services via graphical user interfaces. In most cases, the output data of the services is treated like any other data layer in the system.

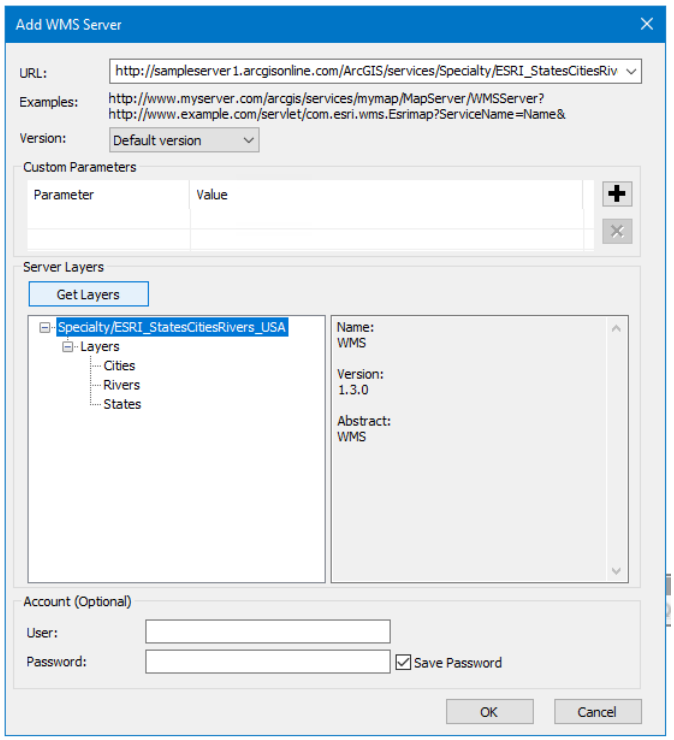
At this point, we would like to take an example of how OGC web services can be integrated into ArcGIS and QGIS.

ArcGIS

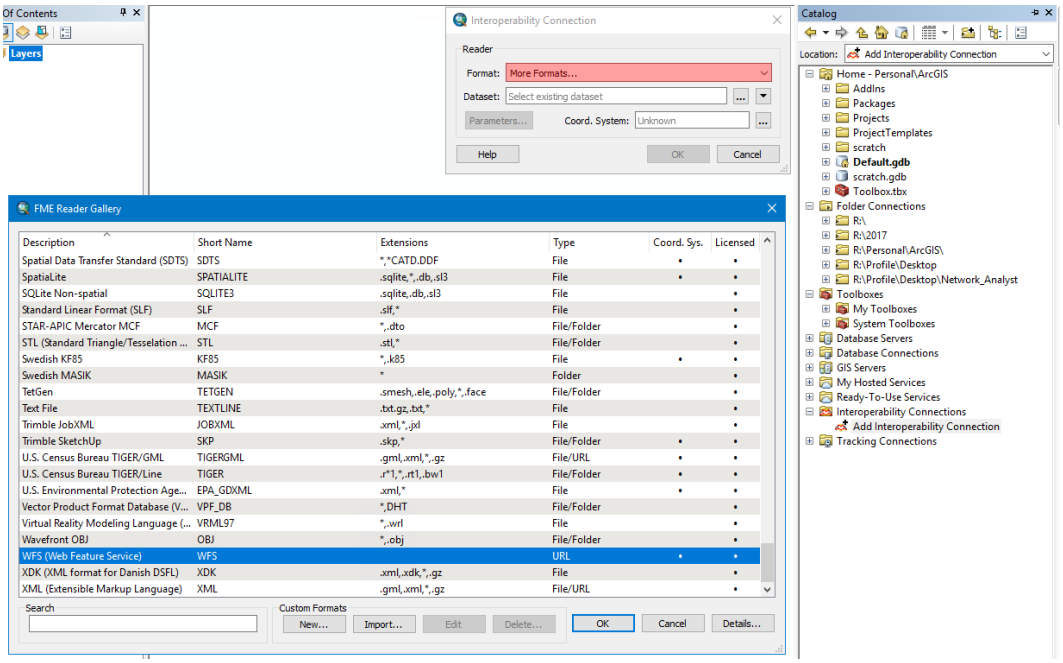
Adding W*S data begins as with all other records: With the "Add Data" button. However, instead of selecting a local folder, we click on the menu item "GIS Servers". The first entries here are ESRI's own server solutions such as ArcGIS Server, followed by the available OGC Services. WM(T)S and WCS are supported natively.



To add a new server, you only need to specify the URL of the service provider. With "Get Layers" you can see directly what map layers/grids are available on the server.



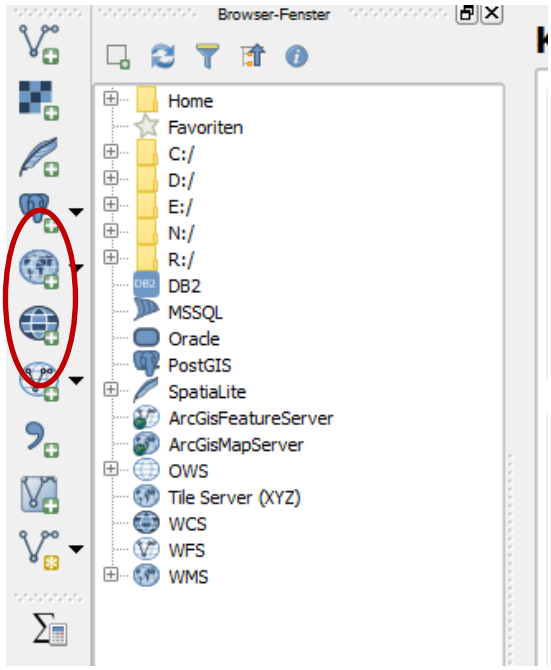
The WFS is not offered at this point. In order to be able to use this ArcGIS must be extended in such a way that it can work with GML data. For this there is the "Data Interoperability" extension. If this is activated, the menu item "Add Interoperability Connection" can be used in the "Catalog" under "Interoperability Connections". This interface can be used to import a large number of geodata formats into ArcGIS. The WFS can also be found here.



This is specified as usual via the URL and can then be used as a layer.

QGIS

QGIS includes support for the same services as ArcGIS, but makes their use even easier for the user. All three options are located in the left menu bar, which also adds all other data to QGIS. In the next menu services can be added as usual via their URL. The user can preview the available layers and choose exactly which data is needed.



Note: In both ArcGIS and QGIS, the interfaces work by appending parameters to the URL given by the user. Nothing is cut off from the URL, i.e. if the user adds certain parameters to the URL, these are typically taken into account in the result. (Unless there are conflicts between the parameters specified by the user and those specified by the program).

Creation of own OGC services

Three things are needed to provide your own OGC service: A web server, an application that implements the OGC standard, and the data to be provided.

Setting up a web server today is easy, and often included in installation instructions for appropriate WebGIS. The server is there to receive and process requests from users on the ports. So the GET and POST requests described in the second section end up on the web server, which passes their parameters and content to locally running applications.

The application is the core of the process. It must take care of interpreting requests, loading data, processing steps, and compiling the response. Typically, this part of the service is taken over by a WebGIS, which can work with functionalities similar to those of a desktop GIS on the web-based data formats. The most popular example is GeoServer, which implements support for almost all major open standards.

The data itself does not necessarily have to be in GML or other formats specified in the OGC standards. However, the respective WebGIS must be able to either provide them directly or convert

them internally into suitable formats. If this functionality is available, data can be delivered from a variety of sources.

Geoserver

If a W*S is not to be integrated into an existing infrastructure, but set up individually, Geoserver makes life very easy for you. First the Java Runtime Environment version 8 (JRE) has to be installed. The installation files provided by Geoserver can then be used for a normal installation. The required web server will be installed in the course of the installation. The ports and interfaces are already set at the end. The exact installation instructions can be found at http://docs.geoserver.org/latest/en/user/.

Literature

Bill, R. (2016): Grundlagen der Geo-Informationssysteme. 6th edition. Wichmann publishing house. Offenbach-Berlin. 867 pages. Chapter 4.3.

Seip, C., Korduan, P., Zehner, M.L. (2017): Web-GIS. Grundlagen, Anwendungen und Implementierungsbeispiele, Wichmann Verlag, Heidelberg, 552 pages.