Gpsinfo Specification Version 2.0

This document defines how gpsinfo online services store data on servers and what API clients shall following in requesting and processing such data. This specification is therefore split into a Server part and a Client part.

Server

This section specifies how gpsinfo stores data on a server. The main idea is to index geospatial raster data in a simple manner that does not require any further functionality from the server apart from serving text-based files. It will be the job of the client libraries to make the data accessible easily. After defining our own server-side configuration file format in version 1 of this specification, we migrated to WMTS in version 2 of this specification. WMTS serves as perfect backbone for our needs.

General

The raster data itself is stored in numerous tiles in ESRI's ASC file format (http://resources.esri.com/help/9.3/arcgisdesktop/com/gp_toolref/spatial_analyst_tools/esri_ascii_raster_format.htm). Gpsinfo's indexing and configuration syntax follows the WMTS standard and amounts to the definitions of an appropriate **gpsinfoWMTSCapabilities.xml** file.

For further information, see the websites of the Open Geospatial Consortium (OGC), the consortium laying down the WMTS standard. The WMTS standard itself is documented extensively**.** The **gpsinfoWMTSCapabilities.xml** file outlined in the following conforms to the standard and is used as template for the *gpsinfo_create* conversion tool.

Examples for WMTS' getCapabilities.xml files may be found at

- http://schemas.opengis.net/wmts/1.0/profiles/wmts-simple/examples/
- http://maps.wien.gv.at/wmts/1.0.0/WMTSCapabilities.xml
- https://www.basemap.at/wmts/1.0.0/WMTSCapabilities.xml

Configuration (gpsInfoWMTSCapabilities.xml)

A gpsinfo service is published and known under its base URL (e.g. http://gpsinfo.org/service). A server may host multiple services with distinct base URLs. Currently, we assume that each service comprises a single layer, defined in its own gpsinfoWMTSCapabilities.xml file. In Appendix A to this document, we share an example XML file. The *gpsinfo_create* conversion tool may be used to generate such an XML file.

Regarding the XML file, see section 7, "WMTS Implementation model" in OpenGIS ® Web Map Tile Service Implementation Standard for tables, what elements are mandatory in the XML file.

Directory Structure

To work around possible limitations of the maximal number of allowed files per subdirectory, the tiles of a layer are organized in subdirectories. Each column is stored in a subdirectory, named by the column's index. The tiles of a column are named by their row index.

http://gpsinfo.org/service/gpsinfoWMTSCapabilities.xml - Configruation containing all layer definitions

http://gpsinfo.org/service/AT_OGD_DHM_LAMB_10M/3/2.asc - Uncompressed tile in ASC file format of tile (2,3).

Clients

Clients are available in different programming languages and realize easy access to the server data. We provide a specification of the client API in pseudo code that follows an object-oriented naming scheme. Arguments to a method are by default input only arguments. Keyword *Out* defines an argument as output only argument (e.g. its value on input is ignored). Keyword *InOut* defines an argument as input and as output argument (e.g. its value on input does matter). All methods except the property getters do return a string that is SUCCESS in case of success and that yields a meaningful error message otherwise.

Server Level - ServerInfo

```
String ServerInfo::connect(String BASEURL)
```

Loads and parses gpsinfo_index.conf from BASEURL.

```
String ServerInfo::baseurl()
```

Returns the BASEURL property.

```
String ServerInfo::version()
```

Returns the VERSION property.

```
StringArray ServerInfo::layers()
```

Returns all LAYERS available from this server.

Layer Level - LayerInfo

```
String LayerInfo::connect(ServerInfo serverInfo, String LAYERNAME)
```

Loads and parses BASEURL/LAYERNAME/gpsinfo_layer.conf.

```
String LayerInfo::layername()
```

Returns the LAYERNAME property. All other properties defined in gpsinfo_layer.conf are accessed accordingly. For the sake of brevetiy, we do not give the full list of getters here.

```
String LayerInfo::value(Method method, Real x, Real y, out Real value)
```

Queries the layer's value at the given coordinates. Method = {CLOSEST, INTERPOLATION}. CLOSEST yields the value of the closest grid point. INTERPOLATION performs bilinear interpolation from the grid cell's corner values. The value is returned in the output argument "value".

```
String LayerInfo::values(
      InOut Real xLowerLeft,
      InOut Real yLowerLeft,
      InOut Real xUpperRight,
      InOut Real yUpperRight,
      Out RealArray values)
```

Queries the layer's values in the interior or on the boundary of the rectangular region defined the given coordinates. On output, the coordinates yield the regions real coordinate bounding box. The values are returned in a row-major array, starting with the lowest line (from left to right).

```
String attribute(Real numericAttribute, Out String stringAttribute)
```

Translates the given numeric attribute into its string representation.

Acknowledgements

In 2019, gpsinfo receives funding from the Internet Privatstiftung Austria (IPA) within the netidee 2018 call.

License Information

Appendix A - gpsinfoWMTSCapabilities.xml (Example)

```
<?xml version="1.0" encoding="UTF-8"?>
<Capabilities xmlns="http://www.opengis.net/wmts/1.0"
xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wmts/1.0
http://schemas.opengis.net/wmts/1.0/wmtsGetCapabilities_response.xsd"
version="1.0.0">
  <ows:ServiceIdentification>
    <ows:Title>gpsinfo Example Layer</ows:Title>
    <ows:Keywords>
      <ows:Keyword>gpsinfo</ows:Keyword>
    </ows:Keywords>
    <ows:ServiceType>OGC WMTS</ows:ServiceType>
```

```
    <ows:ServiceTypeVersion>1.0.0</ows:ServiceTypeVersion>
    <ows:Fees>none</ows:Fees>
    <ows:AccessConstraints>
      https://creativecommons.org/licenses/by/4.0/
    </ows:AccessConstraints>
  </ows:ServiceIdentification>
  <ows:ServiceProvider>
    <ows:ProviderName>Rechenraum GmbH</ows:ProviderName>
    <ows:ProviderSite xlink:href="https://www.rechenraum.com"/>
    <ows:ServiceContact>
      <ows:ContactInfo>
      <ows:Address>
        <ows:City>Vienna</ows:City>
        <ows:Country>Austria</ows:Country>
        <ows:ElectronicMailAddress>
          office@rechenraum.com
        </ows:ElectronicMailAddress>
      </ows:Address>
      </ows:ContactInfo>
    </ows:ServiceContact>
  </ows:ServiceProvider>
  <Contents>
    <Layer>
      <ows:Title>Austria 10x10m DEM</ows:Title>
      <ows:Abstract>gpsinfo's example service for Austria's 10x10
meter OGD DEM (original data (c) geoland.at, released under CC BY
4.0)</ows:Abstract>
      <ows:Identifier>AT_OGD_DHM_LAMB_10M</ows:Identifier>
      <Style isDefault="true"></Style>
      <Format>text/plain</Format>
      <TileMatrixSetLink>
        <TileMatrixSet>AT_OGD_DHM_LAMB_10M</TileMatrixSet>
      </TileMatrixSetLink>
      <ResourceURL format="text/plain" resourceType="tile"
template="http://gpsinfo.org/service/AT_OGD_DHM_LAMB_10M/{TileCol}/{Ti
leRow}.asc"/>
    </Layer>
    <TileMatrixSet>
      <ows:Title>Tiles for layer 'AT_OGD_DHM_LAMB_10M'</ows:Title>
      <ows:Identifier>AT_OGD_DHM_LAMB_10M</ows:Identifier>
      <ows:SupportedCRS>urn:ogc:def:crs:EPSG::31287</ows:SupportedCRS>
      <TileMatrix>
        <ows:Identifier>0</ows:Identifier>
        <ScaleDenominator>35714.285714</ScaleDenominator>
        <TopLeftCorner>500000 503750</TopLeftCorner>
        <TileWidth>190</TileWidth>
        <TileHeight>75</TileHeight>
        <MatrixWidth>5</MatrixWidth>
        <MatrixHeight>4</MatrixHeight>
      </TileMatrix>
    </TileMatrixSet>
  </Contents>
```

```
  <ServiceMetadataURL
xlink:href="http://gpsinfo.org/service/gpsinfoWMTSCapabilities.xml"/>
</Capabilities>
```

Document History

Date	Author	Changes
Feburary, 1 2019	Simon Flöry	First version of server-side spec
February, 26 2019	Simon Flöry	First version of API spec
March, 29 2019	Simon Flöry	Releasing version 1.0alpha of the specification
August, 23 2019	Simon Flöry	Migrating to WMTS
September, 26 2019	Simon Flöry	Version 2