GEOM 4007: Week 4b

Lab logistics, Symbology Encoding, and Filters

Glenn Brauen

Geomatics and Cartographic Research Centre
Department of Geography and Environmental Studies
Carleton University

Jan. 30, 2013

Outline

1. Web (HTTP) server account details and demo

2. Understanding schema definitions: symbology and filters

3. When XML goes bad

4. Notes / Readings

Web (HTTP) server account details and demo

- Set (and remember) a password on the account (now) using `S:\GEOM4008\Putty.exe` to open a Secure Shell (ssh) session:
  - host: `dges.carleton.ca`
  - user name: unique part of your `cmail.carleton.ca` email account.
  - Example: `jimbob@cmail.carleton.ca` would mean the user name is `jimbob`
- Copy files to the web server using Secure File Transfer Protocol (sftp) to connect:
  - same host, login and password as above.
  - specify "sftp" or connect to port 22 (standard for sftp).
  - working folder is automatically set to `/home/jimbob` but this is not where your web files are served from.
  - navigate to your web server folder: `/var/www/courses/GEOM4007/Student/jimbob`
  - put your SLDs there so that remote WMS will be able to access them. Sometimes a few minutes delay is required for changes to be noticed by WMS because they have a cached versions of files.
- Public web address (for file `mySLD.xml` placed in above folder):

  http://dges.carleton.ca/courses/GEOM4007/Student/jimbob/mySLD.xml

Understanding schema definitions

- Good overview of schema provided by *XML Schema Part 0: Primer Second Edition* (W3C 2004)
  `http://www.w3.org/TR/xmlschema-0/#PO`
- See Section 2, “Basic Concepts: The Purchase Order” for an explanation, with examples, of how a schema specifies the structure and syntax for a class of XML documents.

Understanding schema definitions: symbology and filters

Example from OGC (2005, pp. 12–13):

```
<xsd:complexType name="PropertyIsLikeType">
    <xsd:complexContent>
        <xsd:extension base="ogc:ComparisonOpsType">
            <xsd:sequence>
                <xsd:element ref="ogc:PropertyName"/>
                <xsd:element ref="ogc:Literal"/>
            </xsd:sequence>
            <xsd:attribute name="wildCard" type="xsd:string" use="required"/>
            <xsd:attribute name="singleChar" type="xsd:string" use="required"/>
            <xsd:attribute name="escapeChar" type="xsd:string" use="required"/>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
```

Filter I've given you for selecting specific entries from GEONAMES (see `geonames_ottawa_sld.xml` in Assignment #1 templates):

```
<ogc:Filter>
    <ogc:PropertyIsLike wildChar="*" singleChar="." escape="!">
        <ogc:PropertyName>NAME_KEY</ogc:PropertyName>
        <ogc:Literal>OTTAWA</ogc:Literal>
    </ogc:PropertyIsLike>
</ogc:Filter>
```

When XML goes bad

- You will break your XML files.
- Sometimes they will not work for semantic reasons and you may not get any more of an indication of why than a returned empty image. Examples:
  - filter for “OTTAWE” rather than “OTTAWA”.
  - ask for a layer outside it recommended scale.
- Sometimes you will get hints (like the “image not found” icon to the left) and in some of these cases, errors in your XML (Context and / or SLD) may be the cause.
  - Use browser developer tools to fetch just the URL that generated the bad image icon.
  - This will often generate a service exception (XML file) from the WMS.
  - Download and open this file to see the error the WMS is sending you.

When XML goes bad: validate

- In OSGeo Live, once upgraded (hopefully tomorrow), there will be a command line tool called `xmllint` that you can use to validate your XML:
  - `xmllint -format mySLD.xml`
  - will display a formatted version of the file if it is syntactically correct.
  - will display error messages identifying syntax errors otherwise.
  - may be useful to "pipe" command to `more` to allow you to page the contents of the XML file:
    - `xmllint -format mySLD.xml | more`
    - `<space>` pages down
    - `b` pages up
    - `q` exits back to command line prompt
- Online validators. I haven't tried these, but there are a bunch of them if you search. For example:
  - `http://validator.w3.org/`

Notes / Readings

- Monday: project member lists to me **or** statement that you will do an individual project.
- Next lecture:
  - Intro to Mapnik: cartographic renderer.
  - Readings:
    - Discussion of a mapping tool stack, including Mapnik and specifying its role in that stack. (I would draw the tool stack differently). http://alistapart.com/article/takecontrolofyourmaps
    - Discussion of visual processing for web maps and some of the image effects that are being played with for map compositing. http://mapbox.com/blog/expanding-mapnik-carto/
    - Mapnik XML overview. https://github.com/mapnik/mapnik/wiki/XMLConfigReference
    - Mapnik XML example: http://wiki.openstreetmap.org/wiki/Mapnik_Example

References I

OGC (2005). *OpenGIS Filter Encoding Implementation Specification*. Tech. rep. 04-095. Available at `http://www.opengeospatial.org/standards/filter`. Last accessed Jan 2013. Open Geospatial Consortium.

W3C (Oct. 2004). *XML Schema Part 0: Primer Second Edition*. 2nd ed. World Wide Web Consortium. Available at: `http://www.w3.org/TR/xmlschema-0`. Accessed January 2013.