

Web Mapping Services

Best Practices and Implementation Guide for the Federal Geospatial Platform

Document Change Control

| Revision Number | Date of Issue | Author(s) | Brief Description of Change |
|-----------------|---------------|--|--|
| 0.1 | 2014-10-14 | Cindy Mitchell | Creation of first draft for comment |
| 0.2 | 2014-10-17 | Cindy Mitchell | Revisions for scope and topics based on input from Patricia Pollock and Arthur Matson. |
| 1.0 | 2014-12-08 | Cindy Mitchell | Revisions to scope and topics based on input from Patricia Pollock, Eric Boisvert and François Létourneau; division of document divided into two parts to service two end-user audiences (data/web service supplier, data/web service user). This guide will complement the “FGP Geomatics 101” guidance document. |
| 1.1 | 2015-01-15 | Cindy Mitchell | Incorporating content from CGDI Draft Standards Guide in Part II; expanding outline in Part II to include examples, architectural diagrams and generic figures to describe web services operations. |
| Supplier 1.0 | 2015-01-19 | Cindy Mitchell | Revisions based on January 16 meeting with P. Pollock, P. Huppé and Pascale Roussel. For clarity for sharing and review, cut the end-user perspective from this version; will be recreated in a new and separate piece specific for end-user audience. This version specific to internal GoC FGP data providers. |
| Supplier 1.1 | 2015-01-22 | Paul Huppé | Updates to definitions |
| 1.2 | 2015-02-10 | Cindy Mitchell | Addition of web services capability matrix; addition of content and overall consolidation of doc in preparation for complete first draft and community review. |
| 1.3 | 2015-02-11 | Eric Boisvert | First draft review. Results of group discussion and compilation of individual comments. |
| 1.4 | 2015-09-22 | Paul Huppé, JF Bourgon, Marie-Ève Martin, Cindy Mitchell | Added supplemental information on how to customize the html output of a GetFeatureInfo query for both ESRI and MapServer software. |
| 2.0 | 2017-11-01 | C. Mitchell, P. Pollock | Restructured document into four sections; addition of web service quality best practices; addition of multi-layer service guidance. |
| 3.0 | 2019-07-09 | C. Mitchell | General updates, WCAG considerations |
| 3.1 | 2019-07-23 | C. Mitchell | Addition of WCAG Alt Text for images and diagrams. |
| 3.2 | 2019-09-04 | E. Wright | Formatting and WCAG compliance |

Contributors

| Participant | Organization |
|---------------------|---------------------------------------|
| Cindy Mitchell | Natural Resources Canada |
| Diane Lapierre | Natural Resources Canada |
| Brian Low | Natural Resources Canada |
| Chris West | Natural Resources Canada |
| Patricia Pollock | Natural Resources Canada |
| Arthur Matson | Natural Resources Canada |
| Joost Van Ulden | Natural Resources Canada |
| J-F Bourgon | Natural Resources Canada |
| Mike Weech | Environment and Climate Change Canada |
| Mark Shaw | Environment and Climate Change Canada |
| Eric Boisvert | Natural Resources Canada |
| François Létourneau | Natural Resources Canada |
| Paul Huppé | Natural Resources Canada |
| Pascale Roussel | Natural Resources Canada |
| Marie-Éve Martin | Natural Resources Canada |
| Danielle Poirier | Natural Resources Canada |
| Eric Wright | Natural Resources Canada |

Table of Contents

| | |
|--|----|
| Introduction to this Guide..... | 7 |
| Purpose | 7 |
| Audience | 7 |
| Structure | 7 |
| Section 1: Understanding Web Services | 9 |
| What are web services?..... | 9 |
| Web services and interoperability | 10 |
| Geographic web services | 11 |
| How geographic services work | 12 |
| A closer look at Open Geospatial Consortium and Esri web services..... | 13 |
| Discovery services | 13 |
| Web mapping services | 14 |
| Data services | 15 |
| Geoprocessing services | 16 |
| Web mapping service technologies | 16 |
| MapServer..... | 16 |
| Esri..... | 17 |
| OpenGeo | 17 |
| Web mapping application technologies | 17 |
| OpenLayers | 17 |
| Leaflet | 18 |
| ArcGIS API for JavaScript..... | 18 |
| Section 2: FGP Mandatory Requirements for Web Services | 19 |
| The FGP and web services..... | 19 |
| Web map service standards..... | 19 |
| Official languages | 19 |
| Projections | 20 |
| Internet protocol and security | 20 |
| Persistence..... | 20 |

| | |
|---|----|
| Section 3: Best Practices for Quality and Usability | 21 |
| Title | 21 |
| Recommendations | 21 |
| Fees and access constraints | 22 |
| Recommendations | 22 |
| Bounding boxes..... | 22 |
| Recommendations | 22 |
| Attribution..... | 22 |
| Recommendations | 23 |
| Legend..... | 23 |
| Recommendations | 23 |
| Feature attributes | 24 |
| Recommendations | 24 |
| Cartographic representation..... | 24 |
| Colour..... | 24 |
| Symbology..... | 25 |
| Labelling | 27 |
| Scaling | 29 |
| Imagery | 29 |
| Recommendations | 30 |
| Metadata, series and supporting documents..... | 31 |
| Recommendations | 31 |
| Service interoperability..... | 32 |
| Recommendations | 34 |
| Metadata..... | 34 |
| Metadata validation..... | 34 |
| Projections | 35 |
| Styling..... | 35 |
| Performance and Quality of Service (QoS) | 35 |
| Validation and testing | 36 |
| Cross-domain policy for open services | 36 |
| Section 4: How to Create a Web Mapping Service | 37 |

| | |
|--|----|
| Deciding on single or multi-layer services | 37 |
| Publishing a WMS service with GeoServer technology | 40 |
| Create a new workspace | 40 |
| Create a Data Store | 40 |
| Create a Layer | 42 |
| Preview the Layer..... | 43 |
| Publishing a WMS service with Mapserver technology..... | 45 |
| Setup a Mapfile for your WMS | 45 |
| Enabling GetFeatureInfo support | 47 |
| Test your WMS server..... | 47 |
| Sample WMS MapServer Mapfile | 47 |
| Publishing a mapping service with Esri technology | 49 |

Introduction to this Guide

Purpose

The Federal Geospatial Platform (FGP) is based upon a **foundation of operational policy and standards** (OPS) that:

- Supports the **optimal function** of all components of the FGP
- Ensures **compliance** with relevant legislation, policies and standards
- Ensures the **accessibility and interoperability** of Government of Canada geospatial data

This best practices implementation guide supports the FGP policy on Data Structure, which states that web mapping services will be created and made available to support the visualization and simple analysis of FGP datasets. Geographic web services made available through the FGP will meet the minimum requirements of the FGP Data Structure policy; this best practices and implementation guide is a supportive tool for participating departments to aid them in contributing the required web services to the FGP Catalogue.

Audience

The intended audience for this best practices and implementation guide is federal government departments and agencies providing access to metadata, datasets and web services for the Federal Geospatial Platform. The Guide provides introductory information, as well as advanced best practices and instructions, to provide sufficient guidance in the creation of web mapping services regardless of level of expertise of the reader. Table 1 indicates the intended audience of each section in this document.

Structure

This guide consists of four sections, listed and described in Table 1. The Guide addresses everything from “What is a Web Service” to systematic instructions to create an OGC Web Mapping Service or an Esri REST Mapping Service. Readers new to the topic of web services are encouraged to start with Section 1 and work through all subsequent sections; more advanced users can bypass Section 1 and move directly to the content most relevant to their needs.

| Section Title | Scope | Intended Audience |
|---------------------------------------|---|--|
| Section 1: Understanding Web Services | Introduction to the concept of web services, benefits of interoperability and a special focus on geographic web services. | Those new or less familiar with the concept of web services. |

| Section Title | Scope | Intended Audience |
|--|---|---|
| Section 2: FGP Mandatory Requirements | Specific description of mandatory requirements and standards associated with providing web mapping services to the FGP and Open Maps. | Data and metadata custodians from federal departments and agencies providing access to datasets and web services to the FGP and Open Maps. |
| Section 3: Web Services Best Practices for Quality and Usability | Specific description of best practices intended to improve both the quality and the usability of web mapping services in the FGP. | Data and metadata custodians from federal departments and agencies providing access to datasets and web services to the FGP and Open Maps. |
| Section 4: How to Create a Web Mapping Service | Step by step instructions for creating an OGC Web Mapping Service or an Esri REST Mapping Service. | Those who have less or no experience setting up a web mapping service. Note that support for all FGP contributors is readily available via the FGP Data Contributor Support Team. |

Table 1: Sections in this Guide

Section 1: Understanding Web Services

What are web services?

A *web service* differs from a *web site* in that a web service provides information consumable by **software** rather than **humans** – **web services support machine to machine communications**.

At the most basic level, web services are a simple request/response mechanism that allows for remote access and use of data.ⁱ Services communicate directly with each other through a *web client* (the client-side component that asks the web service to return something to them) and a *web server*, passing queries, requests and responses back and forth on the Internet, as illustrated in Figure 1.

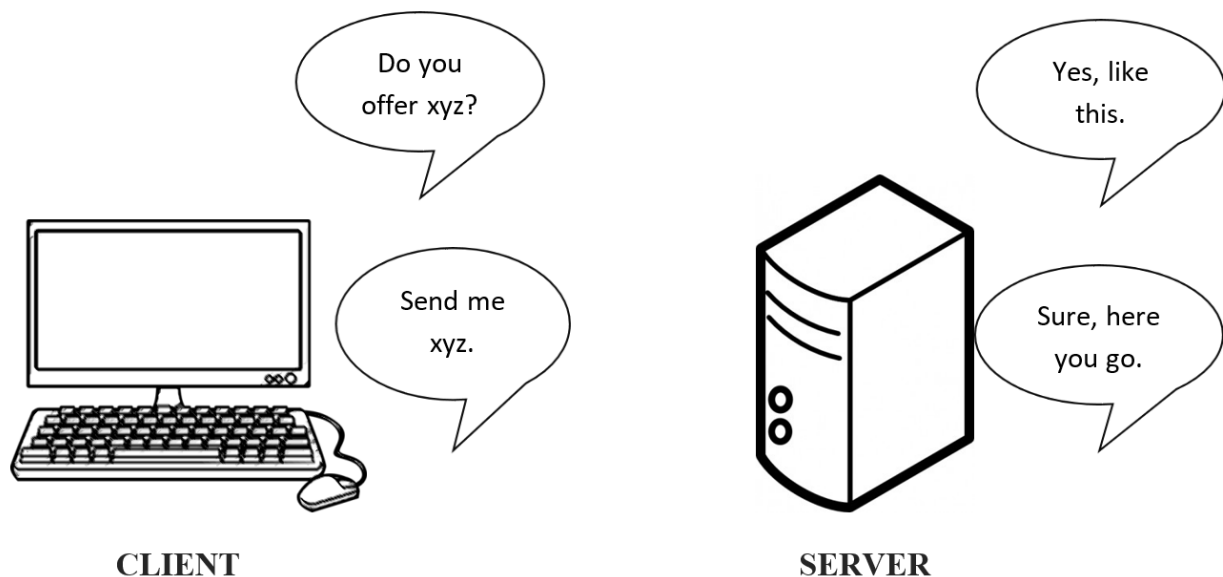


Figure 1: Basic web service server/client communication.

The W3C defines a Web service as **a software system designed to support interoperable machine-to-machine interaction over a networkⁱⁱ**. It is a software function provided at a network address available over the Web.

According to the World Wide Web Consortium (W3C)ⁱⁱⁱ:

- Web services are application components – they are reusable, modular bits of software or other web-based resources that standalone or are loosely coupled to other application components to achieve a series of related functions.
- Web services communicate using open protocols – allowing applications built using different technologies, even proprietary ones, to communicate with each other.
- Web services are self-contained and self-describing – they advertise what they offer, and tell other services how to make use of them.

- Web services are discoverable using Universal Description, Discovery and Integration (UDDI) - a platform-independent directory service for describing, discovering and integrating services using the Internet.
- Web services are used by other applications - they are openly available.

For a more in-depth look at web services, you can access a W3C tutorial at:

http://w3schools.sinsixx.com/webservices/ws_intro.asp.htm

Web services available on the Internet today are typically expressed using JSON (JavaScript Object Notation) or XML (extensible mark-up language). Most client and server frameworks (see Figure 1 again) are designed around using JSON and XML.^{iv} Services are architected according to SOA (service-oriented architecture), or, increasingly, as RESTful (Representational State Transfer) style services.

For more detailed information on JSON, XML, SOA and REST, see Sidebar I and related links.

Web services and interoperability

The key advantage of web services is that they are inherently interoperable. They allow applications built using different technologies to communicate with each other.

A number of standards exist for Web services. Web service standards and specifications further support the interoperability of web services by defining predictable and repeatable means of making the services available on the web.

There are several international standards-setting organizations that produce standards for web services. Web services that deal with geographies are developed by the following organizations:

- International Organization for Standardization (ISO)
- World Wide Web Consortium
- International Hydrographic Organization
- Open Geospatial Consortium

Sidebar I: Encodings and Protocols:

XML: Extensible Markup Language (XML) is a simple, flexible text format derived from SGML (ISO 8879). Originally designed to meet the challenges of large-scale electronic publishing, XML is also playing an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere. See:

<http://www.w3.org/XML/>

<http://www.w3schools.com/xml/>

JSON: **JSON** (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. These properties make JSON an ideal data-interchange language. See:

<http://www.json.org/> and

https://www.w3schools.com/js/js_json_intro.asp

GeoJSON: GeoJSON is a format for encoding a variety of geographic data structures along with their non-geospatial attributes. See:

<http://geojson.org/>

SOA: Services oriented architecture is paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. It provides a uniform means to offer, discover, interact with and use capabilities to produce desired effects consistent with measurable preconditions and expectations. See: www.oasis-open.org

REST: **Representational State Transfer** is an *architecture style* for designing networked applications. The idea is that, rather than using complex mechanisms such as CORBA or SOAP to connect between machines, simple HTTP is used to make calls between machines. RESTful applications use HTTP requests to post data (create and/or update), read data (e.g., make queries), and delete data. See: <http://rest.elkstein.org/>

Geographic web services

Geographic web services function in the same way as any other web service, but deal specifically in the discovery, sharing, visualization, transaction and processing of geospatial data.

Standardization of geospatial web services has been led by the Open Geospatial Consortium (OGC). OGC® standards support interoperable solutions that "geo-enable" the Web. OGC standards empower technology developers to make complex spatial information and services accessible and useful with all kinds of applications^v. The Open Geospatial Consortium (OGC) is an international industry consortium of over 500 private companies, government agencies and universities participating in a consensus process to develop publicly available interface standards. Many market-leading companies, such as Autodesk, DigitalGlobe, Esri, Google and Intergraph are active members in the OGC and their products the OGC's vision of interoperability through the application of common standards^{vi}. Abstract standards for geographic web services produced by the International Organization for Standardization (ISO) are primarily driven by the work accomplished by the OGC.

To learn more about the OGC and its suite of geographic web service standards, please see:

<http://www.opengeospatial.org/standards>

Different types of geographic web services satisfy a range of different requirements (see Figure 2). For example, the purpose of a web map service is to provide a view or picture of the underlying dataset. When requested, a web map service delivers an image in formats like PNG, JPG, GIF or TIFF. A user can request and view the visual representations of one or more datasets simultaneously, which opens up possibilities for layering these images and performing assessment and analysis of the information presented. Ultimately, with a web map service there is a limited ability to query the underlying attributes of the dataset. It is possible to perform a simple query on a web map layer, but this category of service primarily provides a visual representation of the data as a map.

Geographic web services include a sub-class of services known as location-based services (LBS). Location based services (LBS) are services offered through a mobile device using the device's geographical location as the primary driver for services offered. The location of a device is derived from its built-in GPS receiver or short-range positioning beacons, or (less accurately) from the device's location in the mobile network^{vii}. Location-based services are pervasive, if not ubiquitous, on the web, not only driving a huge number of online applications, but contribute enormously to the emerging Internet of Things and the creation of Big Data. Location-based services are the purview of the mobile industry and are not the focus of this guide.

| Service Type | Geographic Data Web Service Capability | | | | | |
|---|--|-----|----------------------|--------|----------|----------------------|
| | Find | Map | Query | Access | Transact | Analyze |
| Discovery Registry service <i>UDDI</i> | ● | | | | | |
| Catalogue service <i>OGC-CSW</i> <i>Esri Search</i> | ● | | | | | |
| Visualize Mapping service <i>OGC-WMS</i> <i>Esri Mapping</i> | | ● | ● <i>(simple)</i> | | | ● <i>(visual)</i> |
| Tile Mapping Service <i>OGC-WMC T</i> | | ● | ● <i>(simple)</i> | | | ● <i>(visual)</i> |
| Data Access Feature Service <i>OGC-WFS</i> <i>Esri Feature</i> | | ● | ● | ● | ● | |
| Coverage Service <i>OGC-WCS</i> <i>Esri Image</i> <i>Esri Globe</i> | | ● | ● | ● | | |
| Tile Coverage Service <i>OGC-WCTS (pending)</i> | | ● | ● | ● | | |
| Data Processing Processing Service <i>OGC-WPS</i> <i>Esri Geoprocessing</i> | | | ● | | ● | ● |

Figure 2: Geographic web services capability matrix.

FGP provides catalogue services, web map services and web map tile services. Future iterations of the FGP will include more advanced services that allow for advanced feature-based queries, access to structured data formats such as JSON and XML, and advanced geospatial processing capabilities to increase the value of the FGP as a tool for policy research and analysis.

How geographic services work

From an end-user perspective, web services and their underlying communications and transactions are transparent. Different services allow end users to find, view, query, access and transact with data.

In a simple human-driven interaction (Figure 3), a person can go to maps.google.com, type in a location, and view a map of that location on their desktop or mobile device. In this instance, the Google user interface calls its own API to return the desired location data and displays it as a map.

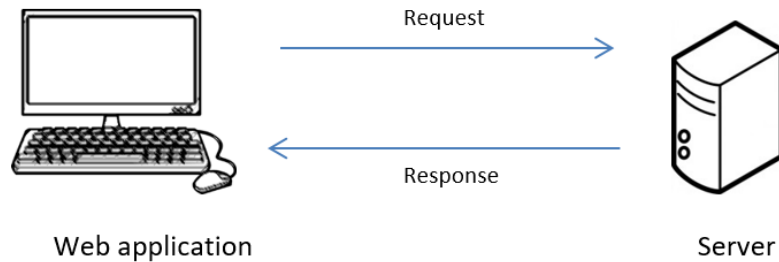


Figure 3: Simple web service request / response

In another example (Figure 4), an organization wishing to embed a map capable of dynamically generating driving directions to their location on their web site could, with a minimum of coding, call a web service that accepts an address and returns a map of the location. .

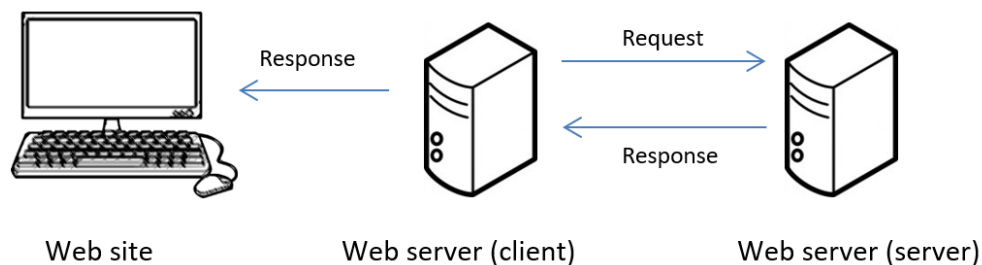


Figure 4: Embedding a web service in an application

Overall, geographic web services are helping to ease the transition from familiar paper maps to electronic maps, and are supporting web-based location services and mobile applications^{viii}.

A closer look at Open Geospatial Consortium and Esri web services

Discovery services

Open Geospatial Consortium – Catalogue Service for the Web (OGC-CSW)

CSW (Catalogue Service for the Web) is an OGC (Open Geospatial Consortium) specification that defines common interfaces to discover, browse, and query metadata about data, services, and other potential resources. Any client supporting CSW (desktop, GIS, web application, client library, etc.) can access and use CSW endpoints (service endpoints are the URL at which the service is available to clients).

Access an online tutorial for CSW at:

<http://cite.opengeospatial.org/pub/cite/files/edu/cat/text/index.html>

Esri Search Service

The Esri search service makes available a searchable index of an organization's GIS content. Search services are most useful in large enterprise deployments where GIS data is spread throughout multiple databases and file shares. Instead of browsing through these data sources, GIS analysts can enter the

URL to a search service and type in some search terms to find data. ArcMap provides an interface for users to search for data in this way and drag and drop resulting datasets into the map^x.

Access an online tutorial for Esri Search at:

<http://resources.arcgis.com/en/help/main/10.1/index.html#/015400000383000000>

Web mapping services

Open Geospatial Consortium Web Mapping Service (OGC:WMS)

The OGC Web Map Service Interface Standard (WMS) provides a simple HTTP interface for requesting geo-registered map images from one or more distributed geospatial databases. The WMS standard defines three operations: one returns service-level metadata; the next returns a map whose geographic and dimensional parameters are well-defined; and an optional third operation returns information about particular locations shown on a map. A WMS request defines the geographic layer(s) and area of interest to be processed. The response to the request is one or more geo-registered map images (returned as JPEG, PNG, etc.) that can be displayed in a browser application. The interface also supports the ability to specify whether the returned images should be transparent so that layers from multiple servers can be combined or not.^x

As the name WMS (Web Mapping Service) implies, the goal of the service is to create cartographic representation and provide a first level of spatial query. Although certain WMS servers can offer more sophisticated features, such as rendering and certain level of querying through SLD^{xi} based styling, the core role of WMS is rendering data. Analysis and download of feature level interaction is beyond its intended purpose and not expected from standard WMS client applications. The interaction model of a WMS client/server pair is to have the client requesting a new map representation every time the view port is changed (zoom and pan, projections, layer lists, image size, etc.). So, navigating using WMS requires pulling a new image from the server when needed and not keeping any data on the client side. This interaction model allows having very complex rendering done by the server without any effort from the client. Since the use case for WMS was to display map in a browser, as a rule of thumb, format that are typically supported by WMS are the ones that can be rendered as is in a browser; PNG, GIF, JPEG, etc.. Multiple layers can be stacked from one or more servers through chaining capabilities. Supporting transparency is critical to achieve this goal.

Learn more about OGC: WMS at:

<http://www.opengeospatial.org/standards/wms>

Esri Mapping Service

NOTE: Many Government of Canada federal departments and agencies currently use Esri ArcGIS as their primary platform for GIS analysis and mapping. Therefore FGP does support and use Esri Mapping Services, along with the open specification WMS.

Esri's map service is the most commonly used ArcGIS service and can contain many capabilities and functions. Esri map services generally support mapping and map viewing, but can also support simple modeling and data analysis. Esri map services can also be publishing using open specifications such as OGC WMS, OGC WCS, and KML.

Learn more about Esri Mapping Services at:

http://resources.arcgis.com/en/help/main/10.1/index.html#/What_is_a_map_service/0154000002m7000000/

Open Geospatial Consortium Web Map Tile Service (OGC:WMTS)

Tile mapping techniques are becoming increasingly popular. A Web Map Tile Service (WMTS) provides access to pre-rendered tiles of geo-referenced data at various scales and projections, not direct access to the data itself. The tile service standard specifies the way in which map tiles are requested by clients, and the ways that servers describe their holdings^{xii}. Tile services are faster because all the work of creating the image is done once and the server's task is reduced at sending the tiles that falls in the extent of the requested map. A potential drawback with tile service is the limited number of projections that it can support (because each projection requires the creation of a large collection of tiles). Tiles are normally used for background layers or imagery.

Learn more about OGC:WMTS at:

<http://www.opengeospatial.org/standards/wmts>

Data services

Open Geospatial Consortium Web Feature Service (OGC:WFS)

Feature services and advanced APIs allow access to the individual features that make up a file. They support complex querying of the data (give me a list of all the roads over 200 miles long with a paved surface) to find and expose only what a user wants. Output formats are the structured information formats (XML, GML, KML, JSON, CSV, and so on) preferred by web application developers.

A Web Feature Service is different from a Web Mapping Service as it serves geospatial data as opposed of a map image. A Web Feature Service operates on feature oriented data. If data is to be selected from multiple sources and fused into a combined data set the feature catalogues from the various data sources need preferably to be in alignment. Schemas, syntax and domain standards, such as WaterML, SensorML and GeoSciML, help to enable interoperability between features.

Learn more at:

<http://www.opengeospatial.org/standards/wfs>

Open Geospatial Consortium Web Coverage Service (OGC:WCS)

A Web Coverage Service is also different from a Web Mapping Service as it serves geospatial data as opposed of a map. A Web Coverage service operates on imagery and other types of coverage data. The standards permit the support of many coverage types.

Learn more at:

<http://www.opengeospatial.org/standards/wcs>

Geoprocessing services

Open Geospatial Consortium Web Processing Service (WPS)

The OGC Web Processing Service (WPS) Interface Standard provides rules for standardizing how inputs and outputs (requests and responses) for invoking geospatial processing services, such as polygon overlay, as a Web service. The WPS standard defines how a client can request the execution of a process, and how the output from the process is handled. It defines an interface that facilitates the publishing of geospatial processes and clients' discovery of and binding to those processes. The data required by the WPS can be delivered across a network or they can be available at the server. WPS can describe any calculation (i.e. process) including all of its inputs and outputs, and trigger its execution as a Web service. WPS supports simultaneous exposure of processes via HTTP GET, HTTP POST, and SOAP, thus allowing the client to choose the most appropriate interface mechanism. The specific processes served up by a WPS implementation are defined by the owner of that implementation. Although WPS was designed to work with spatially referenced data, it can be used with any kind of data. WPS makes it possible to publish, find, and bind to processes in a standardized and thus interoperable fashion. It is transport/platform neutral.

Processing Services can be sequenced into a 'chain' of services to perform specialized processing in support of information production workflows and decision support. Examples include: Coordinate Transformation Services (CTS), Geocoder Services, Route Determination Services, and so on.

Learn more at:

<http://www.opengeospatial.org/standards/wps>

Open Geospatial Consortium Web Coverage Processing Service (OGC:WCPS)

WCPS complements the OGC-WCS by defining a language for retrieval and processing of multi-dimensional geospatial coverages representing sensor, image, or statistics data. Services implementing this language provide access to original or derived sets of geospatial coverage information, in forms that are useful for client-side rendering, input into scientific models, and other client applications^{xiii}.

Web mapping service technologies

MapServer

MapServer is an Open Source geographic data rendering engine written in C. Beyond browsing GIS data, MapServer allows you create "geographic image maps", that is, maps that can direct users to content.

MapServer is a project of OSGeo (<http://www.osgeo.org/>), and is maintained by a growing number of developers (nearing 20) from around the world. It is supported by a diverse group of organizations that fund enhancements and maintenance, and is administered within OSGeo by the MapServer Project Steering Committee made up of developers and other contributors.

Map Server provides support for popular scripting and development environments, including PHP, Python, Perl, Ruby, Java and .NET and operates on Linux, Windows, Mac OS X, Solaris, and more.

Numerous Open Geospatial Consortium (OGC) standards (WMS (client/server), non-transactional WFS (client/server), WCS, Filter Encoding, SLD, GML) are supported by MapServer.

Learn more at:

<http://mapserver.org/>

Esri

Esri (Environmental Systems Research Institute) is an international supplier of Geographic Information System (GIS) software, web GIS and geodatabase management applications. Esri uses the name ArcGIS to refer to its suite of GIS software products, which operate on desktop, server, and mobile platforms. ArcGIS also includes developer products and web services.

Learn more at:

<http://www.esri.com/>

OpenGeo

OpenGeo Suite is a complete geospatial platform for managing data and building maps and applications across web browsers, desktops, and mobile devices. Built on leading open source geospatial software, OpenGeo Suite has a robust and flexible architecture that enables organizations to reliably manage and publish geospatial data.

Open Geo offers the following functionality:

- Manages and store data reliably with PostGIS,
- Publishes maps and data from a variety of formats and sources with GeoServer,
- Builds flexible user interfaces and custom workflows in the browser with OpenLayers,
- Analyzes and publishes geospatial data on the desktop with QGIS,
- Accesses and edits data using open standards,
- Supports OGC standards like WMS, WFS, WCS, and others.

Learn more at:

<http://boundlessgeo.com/solutions/opengeo-suite/>

Web mapping application technologies

OpenLayers

OpenLayers is an open source JavaScript library for displaying map data in web browsers. It provides an API for building rich web-based geographic applications similar to Google Maps and Bing Maps. The library was originally based on the Prototype JavaScript Framework.

OpenLayers supports GeoRSS, KML (Keyhole Markup Language), Geography Markup Language (GML), GeoJSON and map data from any source using OGC-standards as Web Map Service (WMS) or Web Feature Service (WFS).

Learn more at:

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

Leaflet

Leaflet is a widely used open source JavaScript library used to build web mapping applications. Leaflet allows developers without a GIS background to very easily display tiled web maps hosted on a public server. First released in 2011, it supports most mobile and desktop platforms, supporting HTML5 and CSS3. Alongside OpenLayers, and the Google Maps API it is one of the most popular JavaScript mapping libraries, and is now used by major web sites such as FourSquare, Pinterest and Flickr.

Leaflet has core support for the GeoJSON, KML, CSV, WKT, TopoJSON, GPX and WMS formats.

Learn more at:

<http://leafletjs.com/>

ArcGIS API for JavaScript

Esri provides developers with application development framework (ADF) and application programming interfaces (API). The ArcGIS API for JavaScript is a lightweight way to embed maps and tasks in web applications. You can get these maps from ArcGIS Online, your own ArcGIS Server or others' servers.

Learn more at:

<https://developers.arcgis.com/javascript/>

Sidebar II: Commercial and Open Source Software that support WMS:

Commercial:

- Bentley Systems's GIS products
- ESRI's ArcGIS products
- Maptitude Mapping Software
- Cadcorp SIS product suite
- MapInfo Professional
- GeoMedia
- Global Mapper
- Manifold System
- PYXIS WorldView
- Visual Crossing
- StatMap's Stellar Suite products
- Tableau
- MATLAB and Mapping Toolbox
- Netwin
- Autodesk's Map 3D and Civil 3D

Open Source:

- QGIS
- uDig
- OpenJUMP
- MapGuide Open Source
- NASA World Wind
- GRASS GIS
- JOSM
- gvSIG

Section 2: FGP Mandatory Requirements for Web Services

The FGP and web services

The Federal Geospatial Platform (FGP) makes available a collaborative online data visualization and analysis environment that situates the government's economic, social and environmental data in a meaningful spatial context. This environment leverages metadata (standardized records that offer information about the data) geospatial web services and cloud-based mapping and analytical capabilities. In this environment, geospatial data can easily be found, layered and viewed as a map. This capability supports evidence-based decision-making, fosters innovation, and provides better service for Canadians.

To enable these capabilities, the FGP requires that all¹ geospatial datasets be available to view and interact with in this collaborative environment map via an OGC or Esri web mapping service(s). Web map services must comply with standards and best practices that promote interoperability and ease of use.

Web map service standards

The Federal Geospatial Platform is committed to supporting interoperability and the growth of Canada's geospatial data infrastructure. In all cases, departments and agencies are highly encouraged to adopt and implement technical standards that endorsed by the following internationally recognized standards development organizations: the Open Geospatial Consortium (OGC), International Organization for Standardization (ISO) and the World Wide Web Consortium (W3C).

Departments and agencies will make at least one OGC or Esri REST standardized web map service(s) available for each dataset or data collection registered with the FGP. Web map services must comply with either the Open Geospatial Consortium Web Map Service specification (OGC WMS 1.3; ISO 19128:2005; TBS Standard on Geospatial Data), and/or the ESRI REST Mapping Service for ESRI ArcGIS v10.0, or higher.

Official languages

Departments and agencies must provide web mapping services and service metadata in a bilingual format. Metadata for web services must also be provided in a bilingual format. It is highly recommended that two separate web services be created for each dataset, one for each official language. This makes the web service less busy when displayed and easier to view or combine, increasing usability and performance.

¹ In certain cases, it may not possible or feasible to provide a web mapping service for a particular dataset or collection. Refer to Section 4 for examples. Contact the FGP Client Service Desk to seek advice on cases where providing a web service may be difficult. The FGP is equipped to provide solutions for contributors in these cases.

Projections

A map projection is a systematic transformation of the latitudes and longitudes of locations on the surface of a sphere or an ellipsoid into locations on a plane.^[1] Map projections are necessary for creating maps. All map projections distort the surface in some fashion. Depending on the purpose of the map, some distortions are acceptable and others are not; therefore, different map projections exist in order to preserve some properties of the sphere-like body at the expense of other properties. There is no limit to the number of possible map projections, but use of many different projections limit interoperability when layering map services.

Departments and agencies are strongly encouraged to use one or more of the three following data projections for data and web services accessible via the FGP:

- Lambert Conic Conformal EPSG:3979;
- Geographic NAD83 CSRS;
- Web Mercator WGS84 EPSG:3857.

Departments and agencies are strongly encouraged to select a projection that is best suited to the geographic region and / or scale of the dataset. Please note the Web Mercator WGS84 projection is not suitable for areas in the far North.

Internet protocol and security

Departments and agencies will make secure web services available using HTTPS HyperText Transfer Protocol/1.1 hypertext protocol. Servers and clients should be configured to Transport Layer Security Protocol version 1.2 (TLS 1.2) as specified in *RFC 5246 The Transport Layer Security (TLS) Protocol Version 1.2*[9]. Web services that do not conform to HTTPS protocol will not function on servers configured for TLS 1.2. This includes Federal Geospatial Platform and Open Maps servers.

Persistence

Persistent, secure HTTPS URIs will be used to identify FGP online resources, including both dataset access points and web services. This will ensure that all users and applications can dependably access FGP online resources over time.

Section 3: Best Practices for Quality and Usability

This section includes a series of recommended best practices that help generate a solidly consistent user experience. These practices have been shown to promote a meaningful interaction between those view map services and the data those services represent.

Title

Titles are expected to be human readable and comprehensible, and contain little or no jargon, abbreviations, acronyms, etc.. Titles should also be well-formed: short, specific, relevant, providing a user with the ability to readily grasp the content of the service.

A common issue with naming is having very general information, such as an organization name or a data series precedes the name of a specific data subject, pushing the main data subject further to the right and making it harder to read and assess, in various situations.

In other cases this additional name would follow the data subject, making the title excessively long or pushing other information in the title (location, date, scale) further to the right and out of view.

Another issue is the use of unnecessary words in the title, such as “Geographic Distribution” or “Location”. These are generally implied by the fact that the data is being shown on a map.

A common best practice on the web is to limit any title to 70 characters or less, to avoid truncation. Abiding by this rule can also keep titles relatively crisp and relevant. Shorter, simpler names with the most specific information at the front can be beneficial in situations where there is not enough room to display the entire title. It is also important that the title is not overly general. For example, Fisheries and Oceans Canada provides a service titled “Critical Habitat of Species at Risk”, which by name would appear to be a roll-up of all Critical Habitat layers - but it in fact only represents the critical habitat of aquatic species at risk. This is not clear until the product specification is downloaded and read. This title is therefore too general and can be misleading - “Aquatic Species at Risk - Critical Habitat” might be a better choice of title.

Recommendations

Title-1: Use human-readable language as titles and attribute names. Do not use program or other jargon, codes or other alphanumeric strings, and avoid using acronyms and abbreviations.

Title-2: Keep service title length to 70 characters or less, starting with the most unique or important aspect of the data first.

Title-3 : Include enough specificity in the title to clearly identify the subject of the data, avoid also being overly general, suggesting the inclusion of data which is not included.

Fees and access constraints

OWS services allow for the advertisement of 'Fees' and 'AccessConstraints' elements which identify whether there are costs or limitations on access against the entire OWS. While the content model for both of these elements is freetext, OGC standards recommend using the term 'None' (case insensitive) if no fees or access constraints exist and the server wishes to advertise them as such.

Recommendations

Fees-AccessConstraints-1: OWS servers should explicitly advertise Fees and Access Constraints as 'None' in lieu of not declaring these elements in a GetCapabilities response.

Bounding boxes

OGC Web Services typically support a 'GetCapabilities' operation designed to provide a high-level overview (or 'table of contents') of a given OWS which includes a list of layers, feature types, coverages, or other data resource types. A given OWS data resource includes support for advertising a bounding box which represents a minimum bounding rectangle or geospatial extent that represents the OWS data record's geospatial area of interest.

Below are some observations on how OWS server implementations handle this capability:

auto-calculation: the OWS server generates the bounding box value 'on the fly' or automagically. While precise, this could bear a performance cost and overall Quality of Service

global as default: the OWS server provides a default "-180, -90, 180, 90" type value which is representative of the entire globe. This could result in poor search/discovery workflow when assessing the area of interest of an OWS data resource (data encompasses a very small, but important, area of interest)

omission: the OWS server provides no bounding value for an OWS data resource. This results in the user having no knowledge of the spatial characteristics of the OWS data resource

Recommendations

BoundingBox-1: the bounding box should always represent the accurate area of interest of a given resource

BoundingBox-2: if an OWS data resource's bounding box is truly global, the OWS server should advertise as such

Attribution

OGC:WMS (via GetCapabilities) provides a wms:Attribution construct for any given Layer definition. wms:Attribution provides references to the content provider (URL, Title, LogoURL). wms:Attribution provides value for organizational branding (authoritative source) that OGC:WMS clients can use when building a user interface/client.

Recommendations

Attribution-1: WMS server implementations should implement wms:Attribution for the root Layer to provide overall attribution of the WMS server.

Attribution-2: WMS server implementations may implement wms:Attribution for any or all child Layers

Attribution-3: WMS server implementations should ensure wms::Attribution/wms:LogoURL points to a web friendly graphic (PNG, JPEG, etc.), providing an image of smaller dimensions so as to display as a thumbnail by clients

Legend

The legend is perhaps the most intuitive way for the service provider to give the user useful information about the layer.

Depending on the configuration of the service and the type of data being displayed, the legend could be displayed in two different ways: either as a single image including all of the “colour swatches” and all of the text descriptions, or as individual images for each colour swatch accompanied by actual text. When the legend is displayed as a single image, the text descriptions are often difficult to read.

Another problem encountered with some of the legends is that they are too large (i.e. too long, too many attributes or categories) to be displayed in the layer list, without scrolling. Realistically even when the user scrolls down, the large number of categories could compromise their understanding of the display, depending upon the details. Excessively long legend descriptions can also cause the legend to be too wide to be easily viewable.

When the legend includes numeric values, it should also include the appropriate units in the text description of each legend item. Standard International System of Units (SI) abbreviations (i.e. units) should be used in legend descriptions.

Unexplained codes or jargon are not appropriate for a general audience but may be acceptable for expert use.

Recommendations

Legend-1 : The legend should accurately reflect the content of the map.

Legend-2 : A legend should not be provided if the content is not categorized or otherwise has no need of a legend.

Legend-3 : Units, following SI conventions, should be included in the legend descriptive text when measurements are used.

Legend-4 : Codes, contractions or abbreviations should not be used in the legend descriptions if possible, with the exception of SI measurement units and map indexes.

Legend-5 : The legend should be legible. Ensure that method of constructing legend produces a clear, sharp image and easy to read text.

Feature attributes

The assessment of the feature attributes is mostly about readability and understandability.

The most common problem is the use of codes or jargon in either the attribute names or the attribute values.

A minor concern is that floating-point numeric attribute values are often represented at their maximum precision, which can make the numbers difficult to read and falsely represents the actual accuracy of the underlying data.

In some cases point features included the latitude and longitude as attribute values. While this is commonly done as separate attributes, in at least one case the two values were stored in a single attribute as text, separated by a space, which is not as readable for humans or computers. If the data is already being supplied with point geometry, provision of coordinates as attributes really is not necessary.

Recommendations

FeatureAttributes-1 : The use of space-separated words or short phrases for feature attribute names should be used, as opposed to contractions, camelCase or underscores.

FeatureAttributes-2 : The unit of measure in the feature attribute name for measured values should be specified as a separate attribute using SI recognized units. The units used should relate to accuracy and common usage (e.g., the value for the area of a wetlands polygon of 2.34 km² should not be given as 2,338,062 m²).

FeatureAttributes-3 : Numerical precision (i.e., the number of digits) should be given to correspond to and reflect actual accuracy level of the dataset and not to the maximum, machine generated values.

FeatureAttributes-4 : Feature attributes not be included unless they normally have values.

FeatureAttributes-5 : Longitude and latitude should not be given as feature attributes of point data, since equivalent information is contained in the geometry.

Cartographic representation

Colour

Figure 5 shows an example of a map where colours could have been considered more carefully.

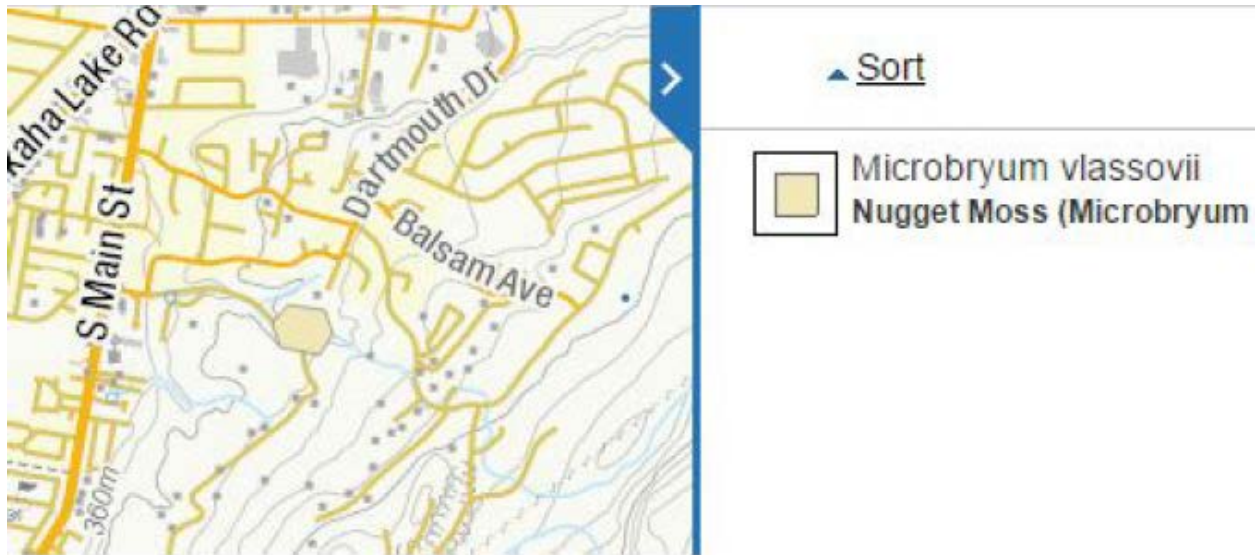


Figure 5: Poor choice of colour makes feature difficult to see against basemap

It is initially difficult to recognize any of the subject features on the map because of the small size of the polygons. Even after zooming in on a specific polygon, it remains difficult to see due to the use of a muted yellowish colour that is similar to both the yellow background and the yellow used for the streets. This colour may have been chosen to differentiate this species from the other species in the series or perhaps simply to avoid garish colours; however, against the default basemap the contrast is much too low.

While the opportunity for colour clashes always exists, one approach might be to use less saturated colours for the basemap, and more saturated colours for the layers of interest. This should at least help increase the visibility a single data layer shown over the basemap. What if the same layer might be used as part of a different basemap? This approach also increases the likelihood of interlayer colour conflicts, because it reduces the colour space available to the non-basemap data layers. These issues will be discussed further in the Service Interoperability section.

Symbology

The “Carbon monoxide emissions by facility” map shown in Figure 6 below displays very effective symbology. The colours are very clear and meaning is enhanced by the different sizes employed. The colours on the legend and map are of different hues, which helps with the user experience. As well, no confusion exists with the background colours or basemap details, although a brighter yellow for 100 to <500 would provide greater contrast with the background. The one real deficiency is the use of gray for No Recordings; it is difficult to see. If No Recordings were symbolized by a gray circle with a heavy black boundary it would work better. The legend could be improved by specifying the time period to which the number of tonnes applies.

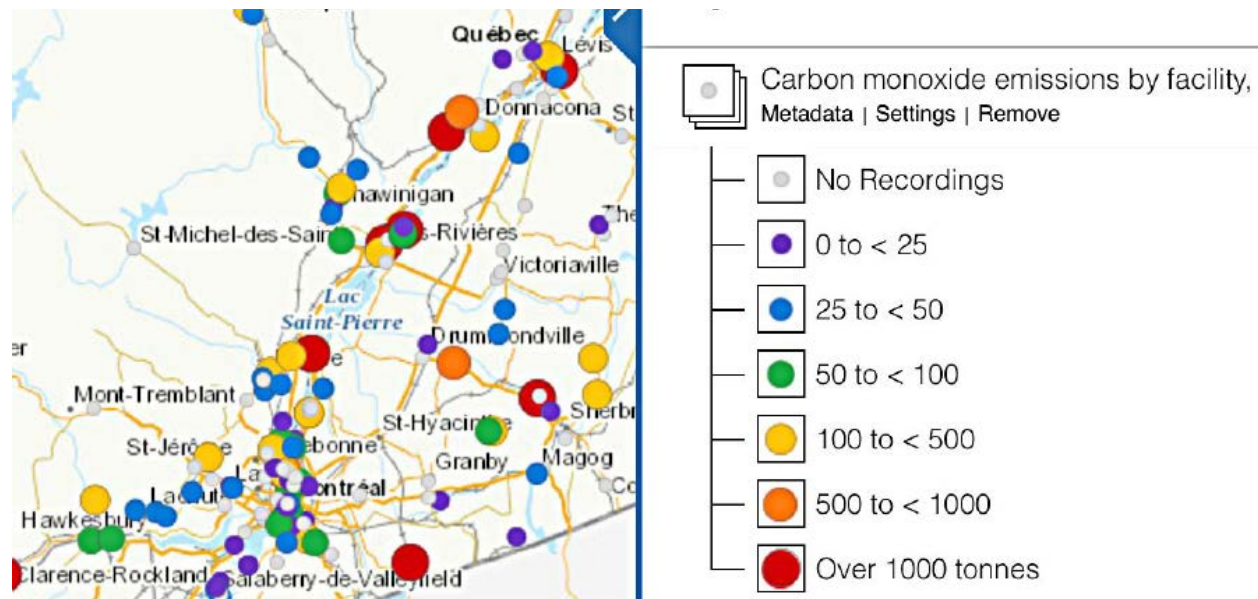


Figure 6: Effective use of colour and size of symbols

“The Major Projects Inventory - Point Geometry” shown in Figure 7 below uses circles, squares, and triangles for different kinds of projects, as shown in the large image below. The colours are sufficiently different to aid recognition, as are the different sizes of the symbols. Also, note that the darker tones in all three cases are used for Planned, whereas lighter tones are employed for Under Construction. The black boundary used to indicate the largest planned and under construction sites is used on only a single site across the country (shown on the map snippet on the left).

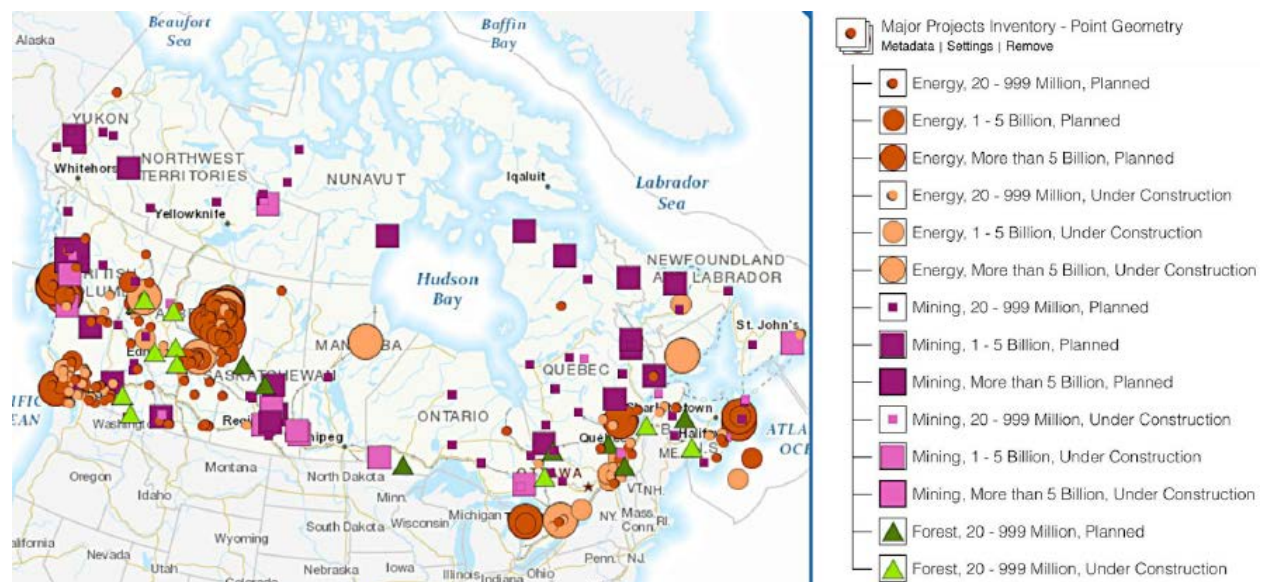


Figure 7: Effective use of colour, shape and size of symbols

This raises the question of whether the legend should contain symbols that cannot be found anywhere in the country, as is the case with Mining, More than 5 billion, Under Construction. On the other hand, by including the category it is clear in this case that nothing has been forgotten.

The legend should also indicate the units used, which will differ depending upon the subject.

The “Metal Mines - Producing Mines” map in Figure 8 shows symbols with a flat design, characterized by no use of gradients, textures or drop shadows. Not only is this in line with modern design, as used on smartphones for example, but it also makes displaying the symbols more compatible with different map rendering technologies.



Figure 8: Simple flat design for symbols

Labelling

Figure 9 provides an example of good use of labelling. On the left are the NTS 1:250 000 blocks shown with labels and 1:50 000 subdivisions. Zooming in further (on the right) shows the labels for the 1:50 000 mapsheets. The labels are clear and the overlay against the basemap is quite well done in both cases. Zooming further (not shown) shows the blocks and the labels for the 1:20 000 NTS grid.



Figure 9: Clearly labelled blocks on NTS maps at different zoom levels

In the Historic Treaties map shown in Figure 10 below, labels and different colours would do much to improve the useful information content of this map. The various green colours are difficult to identify with certainty with the listing in the legend. So this service has issues with labels, colours, legend content, and legend length.

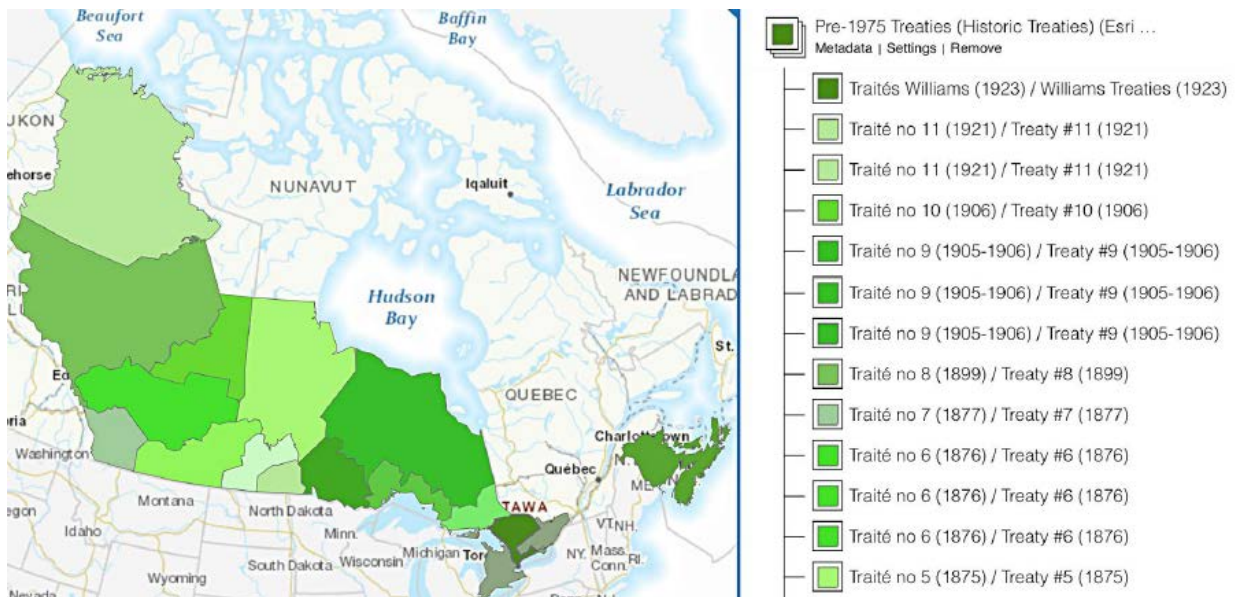


Figure 10: Use of labels would improve usability

In the case of feature level services, some viewers display a feature “name” when a feature is hovered-over with the mouse cursor. It’s not uncommon to find when a feature “name” is displayed when hovering, it is a code or jargon of some sort, where a long-form name is actually available in the data. Since this form of labeling does not take up permanent space on the map image, it seems there is no reason not to use the most descriptive name.

Scaling

For a general use web service, it is important that something be clearly visible on the map at all resolutions. In the case of small polygon data, or small raster areas, it is often invisible on the map at a smaller (i.e. national or provincial) zoom level, and would better be represented as point symbols or a polygon area to help a user navigate to where the data is and zoom in on it.

Conversely, it should not be possible to request more data than can be returned in a reasonable period of time; when zoomed out, either more general data should be displayed or a point symbol used. For feature-level services, expert users should still be able to access the full detail data.

Imagery

This example shown in Figure 11 from the “Swift Current LiDAR Project 2009 - Orthos” layer shows some edge artifacts likely caused by re-tiling or re-projecting the ortho image tiles. This sort of problem may affect the usability of the data from an analytical perspective. Of more importance here is that it is an assault on the viewer’s sensibilities and should be easily avoidable with due care taken in processing the data.

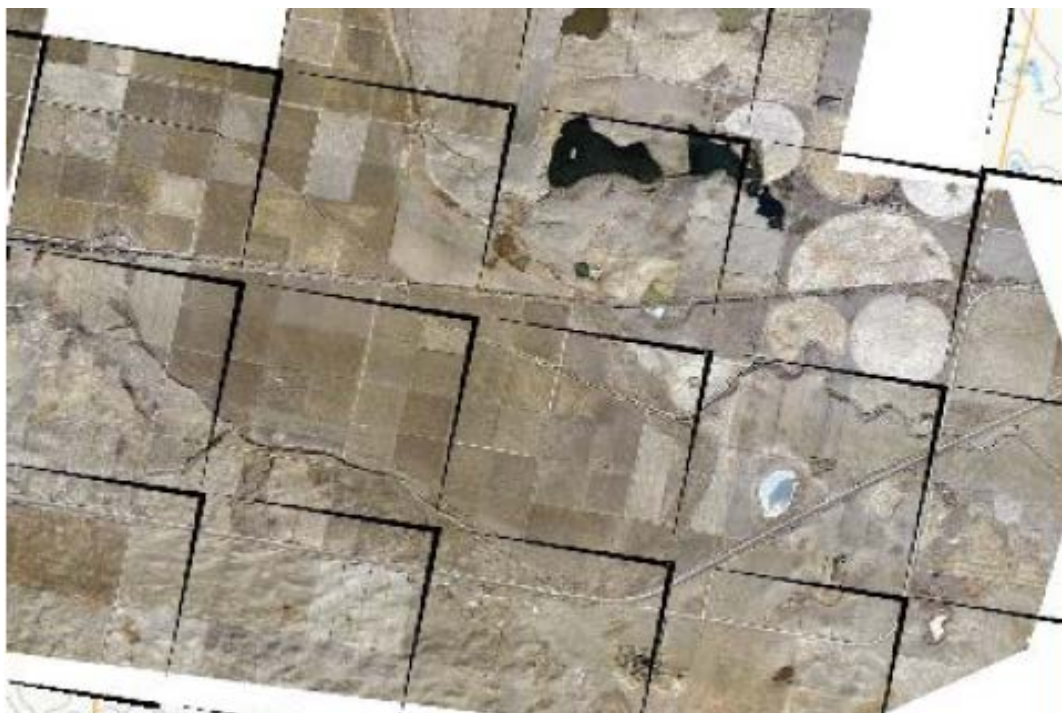


Figure 11: Edge artifacts generated in processing create a choppy look and impede usability of the mosaic

Another point about lidar imagery concerns their meaning to the typical user. Such imagery is often shown as shades of gray with the darker sections showing lower elevations and the lighter sections displaying higher elevations. This should be indicated through the legend preferably and otherwise by the metadata. Even more useful would be to show the range of elevation in the area of question. The experienced user may realize that the the dark shades represent lower areas, but will not necessarily

know if the difference between these lower areas and the higher elevation areas is centimetres, metres, tens of metres, etc.

Recommendations

CartographyColour-1 : Colours for features/layers should be chosen that are in clear contrast with those on the basemap.

CartographyColour-2 : Colours appearing on individual layers should be readily distinguishable from one another, unless by intention they are of the same colour.

CartographyColour-3 : If it is known that certain layers are likely to be used in combination with one another, then care should be taken to ensure that similar colours are not used on the different layers.

CartographySymbology-1 : Symbols should have colours that contrast sufficiently with the basemap details and with one another. Subdued or pastel colours should be avoided.

CartographySymbology-2 : If the symbols include contrasting boundaries used to distinguish symbols from one another, then the boundaries should be comparatively thick so that they are easily discerned on different devices and screens of differing qualities.

CartographySymbology-3 : The use of different sizes and colours in combination is recommended for rendering different numeric categories.

CartographySymbology-4 : The use of different shapes, such as triangles, squares, and circles, is recommended for portraying different series on the same map.

CartographySymbology-5 : Connotative symbols with varying shapes or internal icons can be used so long as colour is also used to distinguish them.

CartographySymbology-6 : All symbols should have a flat design, without the use of gradients, textures, or drop shadows.

CartographyLabelling-1 : If used, labels should be short and readily understood directly or from the legend, with the exception of the label as a map index, in which case it is acceptable if the explanation is found in the metadata.

CartographyLabelling-2 : Where large polygonal features are displayed, labels are recommended if practical to implement. For small features on the map, care must be taken that the label does not conflict with other labels or with boundaries.

CartographyLabelling-3: In cases where feature name is configured to display when hovered over, avoid providing codes, acronyms or other cryptic text if a long-form name is available in the data.

CartographyScaling-1 : Services should not allow requests for excessive amounts of data that would cause the server or connection to timeout. If a service provides only high-resolution data, it should only be available at large scales.

CartographyScaling-2 : Wherever possible, lower resolution data or alternative representations (e.g. a point symbol instead of a set of lines or polygons) should be provided at smaller scales to enable the user to navigate the map to the data of interest.

CartographyImagery-1 : Quality assurance should be carried out so that a proper orthomosaic is available, without obvious artifacts.

CartographyImagery-2 : When a color gradient is used to visualize the data, such as a greyscale map of elevation data, the legend should indicate the values associated with key colors in the gradient, eg. that the lighter and darker areas represent higher and lower elevations, respectively.

Metadata, series and supporting documents

The service “abstract” should contain at least a short overview describing the dataset. In some cases, the provided abstract provides no more information than the title of service.

Ideally, a link to a “product specification” or similar document should be provided if one exists. A suggested standard for the minimum information that should be provided in such a document could be:

- A data dictionary for codes or terminology
- Description of the methodology for the creation/capture of the data
- Reference to any related laws or standards

The name of the “data series” that a service belongs to should be recorded in the service metadata, searchable and displayed to the user through the map viewer interface. The map viewer’s “metadata” display should include the name of the data series, links to view the metadata of the other layers in the same series, and links to add some or all of those other layers to the map.

Recommendations

MetadataSeries&SupportingDocs-1 : Every service should include an abstract with meaningful content. The content should include more detail than the title, so that in a few sentences the reader has a fair idea as to what the service provides.

MetadataSeries&SupportingDocs-2 : Any documents intended for a general audience should minimize the use of jargon and abbreviations. If such terms are commonplace or judged to be unavoidable, they should be briefly defined.

MetadataSeries&SupportingDocs-3 : It is strongly encouraged that service metadata include a link to a “product specification” or similar document. Such document(s) should include as a minimum a data dictionary, a description of the creation/capture methodology, and references to any related laws or standards.

MetadataSeries&SupportingDocs-4 : Ensure support for the concept of “data series” in metadata records.

Service interoperability

Issues involving service interoperability are often identified during a quality of experience assessment. In particular, there are frequently difficulties with:

- Visualization of a service displayed over a basemap
- Visualization of multiple services when displayed together
- Querying two similar services

An example of poor visualization of a service over a basemap is shown in Figure 12 below, from the service “Sidney Island Shorebird Surveys - Transects Line”.

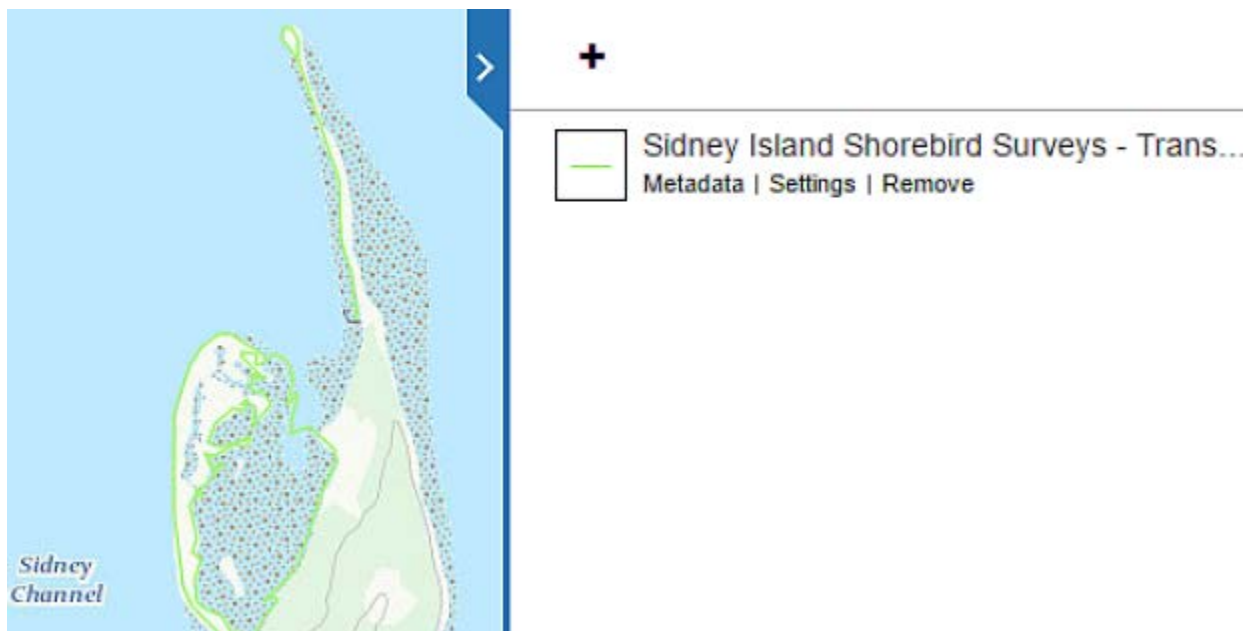


Figure 12: Vector layer is nearly invisible over base layer due to poor choice of colour and line weight

The thin light green line is almost invisible against the background. Visibility would be improved with a thicker line of a more contrasting colour. With so much green on the basemap, the colour of the line should have a different hue altogether. In the current case the line is so thin and light that even tested against a different coloured basemap, such as a shaded relief option, its rendering needs improvement.

Figure 13 shows the difficulty in visualizing and querying two services together. In this case, several layers in a related series of data (Canadian Environmental Sustainability Indicators (CESI)) were displayed on the map together. These layers use the same symbology of coloured points: increasing in size and tending toward red as they get larger (worse). This results in a very readable map, individually. However, when multiple CESI layers are loaded at the same time, it is impossible to determine which symbol relates to which dataset, just by viewing the map. Furthermore, querying the map shows that the two data sets have nearly identical attributes. It is difficult to recognize which layer's feature has been selected, because the map client used in this case did not identify from which layer the results

were returned. Looking carefully at the attributes of the feature helps in this case because the attribute name “Sox” corresponds to the name of the pollutant. In a worse case, it is possible that multiple datasets could share identical attribution and styling, and be indifferentiable when queried.

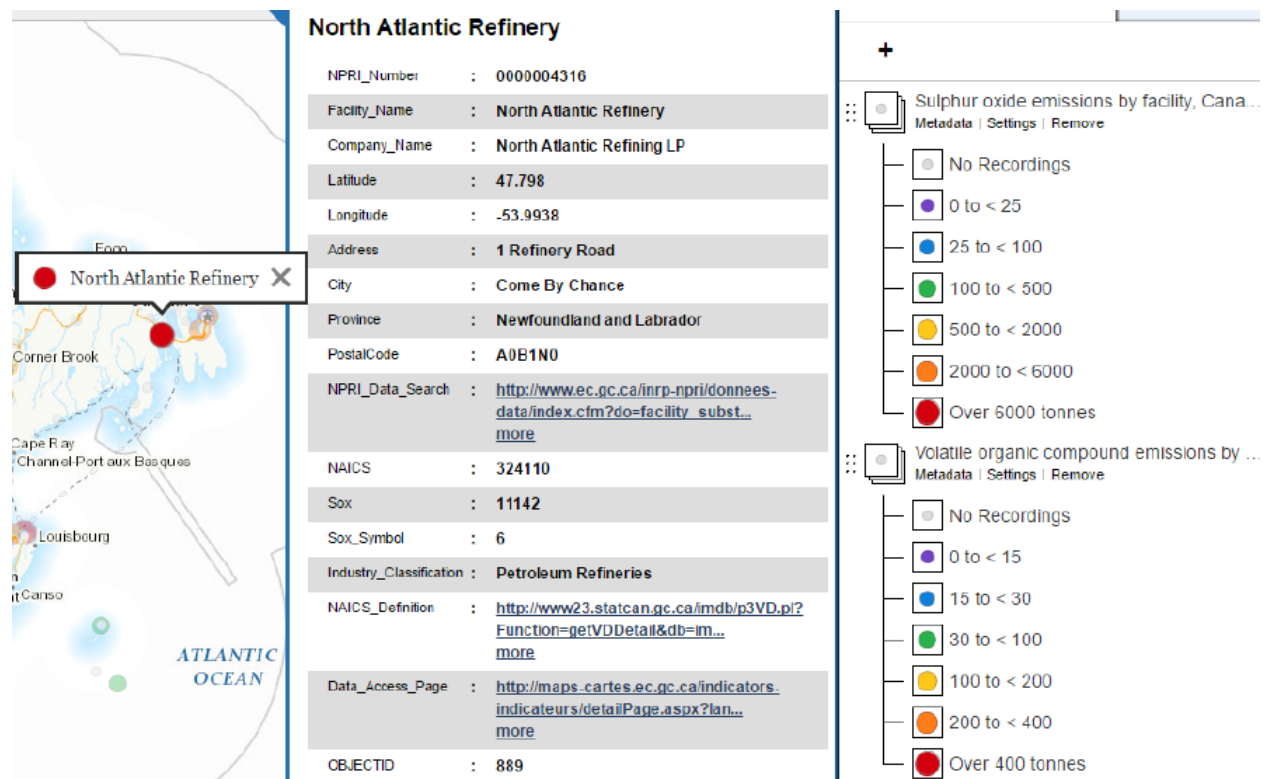


Figure 13: Displaying two services together can create unexpected usability issues for the viewer

The way around these problems is to be able change the styling of a service (colors, linetypes, and/or symbols) dynamically, depending on the situation. Three relevant solutions to this problem are recognized:

- WMS Named styles
- WMS Styled Layer Descriptor (SLD) support
- Client-side style definition and rendering of layers

The WMS specification (all versions since 1.0.0) allows for each layer to support zero or more named styles, with the names of the styles (and other descriptive information) specified in the layer’s capabilities in the WMS server’s capabilities document. This allows the service provider to provide more than one style for a given layer, and requires only minimal support on the client side (allowing the user to select from a list of style names).

The WMS specification also provides for optional support of the SLD specification, which includes the ability for the client to provide an SLD document describing how to style the layer, with a getMap

request. This allows the client to specify the style information in the SLD, and have the server render the map according to that SLD. This implies a sophisticated client application, which provides a user interface to define the style, generates the appropriate SLD XML, and makes use of an XML-based HTTP POST getMap request to send the SLD to the server (instead of the typical HTTP GET getMap request). While this provides excellent flexibility and configurability, supporting it on the server side allows the user to specify arbitrarily complex styling that could cause excessive workload for the server. To reduce the level of sophistication required in the client, another approach is for the service provider to provide additionally, alternate SLDs. These alternate SLDs would work similarly to named styles, except instead of being defined in the server configuration, and referenced by name, they would be defined by a separate SLD file and reference using a URL. It is even possible for a third party to define and provide the SLD, a job that could be filled potentially by an FGP styling team. This requires some basic support in the client and a way to communicate the availability of the alternate SLDs to the client, likely through some sort of metadata.

Where the service sends feature-level data, not map images, the client renders them to a map image using the styling suggested by the service. Some software supports complete restyling of the data on the client-side, and so can a sufficiently sophisticated web-map client. This provides the greatest flexibility, but requires that both the server and the client handle the individual features and coordinates being rendered on the map, which in some cases is a significant overhead, even preventing some layers from loading or displaying properly because of the volume of data.

Recommendations

ServiceInteroperability-1 : Implement support in web-mapping clients for WMS named styles, and recommend that WMS services offer more than one named style. In the case of data series, one style should be visually distinguishable from other data in the series, and another style should be similar or identical to other data in the series.

ServiceInteroperability-2 : Investigate the compatibility of different services with SLD. It should be determined if a single SLD can be published and used correctly by WMS servers from different vendors (Mapserver, Geoserver, ArcGIS Server). Further investigation will be required to determine who should be responsible for producing and maintaining such SLDs, and how they can be discovered and used.

Metadata

Metadata is the documentation that describes data, services and applications. In human-readable form, it has primarily been used as information to enable a user to understand, compare and interchange the content of the described resource. In the Web Services context, XML-encoded (machine-readable and human-readable) metadata stored in catalogs and registries enables services to use those catalogs and registries to find data and services.

Metadata validation

A Metadata validator will help to ensure the consistency of the submitted information and that the information meets the requirement set out by the Metadata schema. Validators range from simple XML validators to more complicated tools that check code lists, coordinates, projections, and dimensions,

size and extents of the metadata to the actual spatial data. It is important that the Metadata be properly validated as consumers of the Metadata will heavily rely on this for resource information.

Projections

Commonly a **map projection** is a systematic transformation of the latitudes and longitudes of locations on the [surface](#) of a [sphere](#) or an [ellipsoid](#) into locations on a [plane](#).^[1] Map projections are necessary for creating [maps](#). All map projections distort the surface in some fashion. Depending on the purpose of the map, some distortions are acceptable and others are not; therefore different map projections exist in order to preserve some properties of the sphere-like body at the expense of other properties. There is no limit to the number of possible map projections

Styling

The OGC Styled Layer Descriptor (SLD) Profile of the OGC Web Map Service (WMS) Standard defines an encoding that extends the WMS standard to allow user-defined symbolization and coloring of geographic feature and coverage data. SLD addresses the need for users and software to be able to control the visual portrayal of the geospatial data. The ability to define styling rules requires a styling language that the client and server can both understand.

The OGC is currently investigating the extension of the SLD standard to include 3D, thematic and other extensions.

To learn more about SLD for WMS, see:

<http://www.opengeospatial.org/standards/sld>

The OGC Symbology Encoding (SE) provides an XML language for styling information that can be applied to digital Feature and Coverage data as well as Web Map Services.

To learn more about symbology encoding, see:

<http://www.opengeospatial.org/standards/se>

Performance and Quality of Service (QoS)

Performance and Quality of Service is important when Web services are so prolific and inherent in distributed systems. Delivering QoS over the Internet is a very important challenge. With the distributed nature of the internet and the vast distances of consumers of the data and services, Web products must address performance, availability, reliability as well as security.

Performance is typically how well the Web services are, and can be measured in terms of throughput and latency (how long it takes to perform its tasks). High throughput and low latency results typically reflect a good performing Web service.

Availability is the quality of a Web service that reflects if the service is ready for use when being requested. A probability that a service is available is typically measurement of Availability. A large availability value would translate to a high readiness where a low value would typically represent a unpredictable service.

Reliability reflects how a Web service is maintained and the service quality. For example the number of failures over a given time period (example the number of failures per week) are used to evaluate how reliable a service is. It is important to understand this indicator and how it relates to the actual service. Many Web services are time or even seasonal sensitive. For example some services are only used during work hours, or some our seasonal (fire season).

Security is an essential component in a Web services Quality of Service and performance indicator. This indicator reflects how confidence and authenticity of Web service are measured. Indicators of security may include the use of encryption methods, access control, authentication, and authorized, authentic data source.

These and other indicators help systems developers and maintenance teams to deliver the best Web Services to consumers. Historical and current indicators can be used to help adjust systems and services to better improve performance and Quality of Service.

Validation and testing

Like metadata, Web services can be validated. Validating Web services increases interoperability and allows for the reduction of integration risks. Agencies and organizations gain confidence knowing that their Web services are interoperable and available to consumers in a standardized mechanism.

The OGC has a free online testing facility available to the public at:

<http://cite.opengeospatial.org/teamengine/>

Cross-domain policy for open services

To enable open access across domain boundaries to web services that have been made publically available, it is strongly recommended that contributors to FGP open their web services servers by adding the following header to the server's response:

```
Access-Control-Allow-Origin: *
```

To learn more about implementing this on a specific platform, please see: <http://enable-cors.org/server.html>

Section 4: How to Create a Web Mapping Service

Deciding on single or multi-layer services

To meet requirements of all user groups, fulfill all FGP objectives, and support all possible applications of FGP data and services, data contributors should make as many resources available as possible, and as openly as possible.

In the first release of FGP, the “one dataset, one service” paradigm was used. One single-layer web service was made available for each dataset registered on the FGP.

Now, the FGP supports a more comprehensive paradigm, where the contributor may choose how to best organize and share geospatial data. It is possible to create a relationship between a collection of geospatial data (the parent) and the individual products (children) that make up the collection. A multi-layer service can be associated with a collection, or each product may have its own service, with one or more layers included there.

The decision on how to make geospatial services available is up to the data contributor/custodian. FGP recommends, however, that to meet requirements of all user groups, fulfill all FGP objectives, and support all possible applications of FGP data and services, data contributors should make as many resources available as possible, and as openly as possible.

Scenarios, examples and recommendations shown in Table 2, describe common dataset presentations and are intended to support decision-making on how and when to use a single or multi-layer service. This list is not exhaustive and is provided as a guideline only. The FGP Contributor Support (NRCan.fgp-pgf.RNCan@Canada.ca) team can offer advice should this information not provide sufficient guidance for your particular scenario.

| | Scenario | Example | Recommendation |
|---|---|---|---|
| 1 | Data collection with child datasets | Collection : Critical Habitat for Species at Risk - British Columbia Related Datasets : 52 species at risk | When the children can be shown individually, A collection record with web service and A dataset record with web service per child |
| 2 | Dataset with layer for each geometry type | Dataset A : Northern major projects - Line Geometry Dataset B : Northern major projects - Point Geometry | Line and Point geometry must be offered (viewable, downloadable) together, under the same record. When the components are indissociable, One dataset record with one service (Multi/Group) |

| | Scenario | Example | Recommendation |
|---|---|--|--|
| 3 | Multiple classes with hierarchy between classes | <p>Dataset : Surficial Geology. <i>All classes makes a whole. Most classes make no sense in isolation while some does.</i></p> <p>Collection/Dataset: North American Cooperation on Energy Information <i>(3 types of border crossings, 3 types of refineries, 11 types of power plants 100mw and over, ...)</i></p> <p>All classes makes a whole, but some sub classes could be very useful solely.</p> | <p>One dataset with one service (Multi/Group), or, One data collection (with service) and child dataset records (with service)</p> |
| 4 | Time Series | <p>Case 3a - One event with repetitive coverage => Flood , Earthquake A data set for the time series with a web service composed of all the component</p> <p>Case 3b - Electoral boundaries dataset at every federal election (each election)</p> <p>Case 3c - Community well being – re-evaluation every 4 years</p> | <ul style="list-style-type: none"> • A dataset record and one web service (multi/group layers) for each events or series of events <p>OR</p> <ul style="list-style-type: none"> • A collection of web services for all the events and a dataset record per event (Childs), each having a web service <p>OR</p> <ul style="list-style-type: none"> • A dataset with the web service showing the actual data with obsolete data in download |
| 5 | Tiled Data product | <p>Dataset : Canadian Digital Elevation Model. The product is available tiled (by NTS, by province, by project, etc) and all the data sets have the save data structure. Data are generally tiled because of size.</p> | <p>When the data is tiled and all tiles have the same structure, One dataset and one service</p> |

| | Scenario | Example | Recommendation |
|---|---|--|--|
| 6 | Representation related to scale | <p>Dataset A: Community well being is provided as a multi-layer service. Four layers are provided in total. 2011 census data is expressed as centroids (points better for viewing at small scale) and boundaries (polygons better for viewing at larger scales).</p> <p>Dataset B : Road network (with many feature classes) showing a selection of features at small scale, and the complete information at large scale. For dense data, features selection depending on the scale is generally required.</p> | <p>Case 3 (collection/datasets or dataset)</p> <p>Consideration for scale dependency when viewing through provision of multiple layers (Dataset A); and use of scale dependency when rendering service (Dataset B)</p> |
| 7 | Same data with multiple representations | <p>Dataset A : Socio economic data – sector diagrams</p> <p>Dataset B: Socio economic data – choropleth (shaded polygons)</p> <p>A dataset could be presented with different symbology depending on the user needs</p> | <p>When more then one web services show the same data, both web services are register to the same record. Only one of them can be associated to the button View on map. The second web service will have to be register using a different protocol (or in the data resource section with the protocol HTTPS).</p> |
| 8 | Dataset without web service | Important geospatial may be registered, as long as web service and/or download is available. Often, archived geospatial data don't have web service, but have download data still valuable. | A record may be registered for a dataset even if no web service exists. However, the download is mandatory. |

Table 2: Scenarios, examples and recommendations to guide creation of single or multi-layer service

Publishing a WMS service with GeoServer technology

The proposed procedure is derived from information available online at the following location:

<http://docs.geoserver.org/latest/en/user/gettingstarted/shapefile-quickstart/index.html>

Create a new workspace

Within a web browser navigate to <http://yourhost:8080/geoserver/web> and log in using administrative account. Then navigate to Data -> Workspaces

Workspaces

Manage GeoServer workspaces

 Add new workspace

 Remove selected workspace(s)

<< < 1 > >> Results 1 to 7 (out of 7 items)

 Search

| <input type="checkbox"/> Workspace Name |
|--|
| <input type="checkbox"/> sf |
| <input type="checkbox"/> topp |
| <input type="checkbox"/> it.geosolutions |
| <input type="checkbox"/> sde |
| <input type="checkbox"/> nunc |
| <input type="checkbox"/> tiger |
| <input type="checkbox"/> cite |

<< < 1 > >> Results 1 to 7 (out of 7 items)

Create a new workspace by clicking the *Add new workspace* link. You will be prompted to enter a workspace Name and Namespace URI. Enter the desired name for your new workspace and define the associated namespace then click *Submit*.

New Workspace

Configure a new workspace

Name

Namespace URI

The namespace uri associated with this workspace






Create a Data Store

Navigate to Data -> Stores then click on the *Add new store* link. You will be redirected to a list of the data sources supported by GeoServer.






New data source

Choose the type of data source you wish to configure

Vector Data Sources

-  [Directory of spatial files](#) - Takes a directory of spatial data files and exposes it as a data store
-  [PostGIS NG](#) - PostGIS Database
-  [PostGIS NG \(JNDI\)](#) - PostGIS Database (JNDI)
-  [Properties](#) - Allows access to Java Property files containing Feature information
-  [Shapefile](#) - ESRI(tm) Shapefiles (*.shp)
-  [Web Feature Server](#) - The WFSDataStore represents a connection to a Web Feature Server. This connection provides access to the Features published by the server, and the ability to perform transactions on the server (when supported / allowed).

Raster Data Sources

-  [ArcGrid](#) - Arc Grid Coverage Format
-  [GeoTIFF](#) - Tagged Image File Format with Geographic information
-  [Gtopo30](#) - Gtopo30 Coverage Format
-  [ImageMosaic](#) - Image mosaicking plugin
-  [WorldImage](#) - A raster file accompanied by a spatial data file

Select the appropriate file format. Assuming your data is available as a Shapefile, you click on the *Shapefile - ESRI(tm) Shapefiles (.shp)* link. The New Vector Data Source page will display.

Select your newly created workspace from the drop down menu. Enter the *Data Source Name* as and provide a brief *Description*.

Under Connection Parameters specify the location URL of the Shapefile as
file:///path/to/your/shapefile.shp

New Vector Data Source

Shapefile
ESRI(tm) Shapefiles (*.shp)

Basic Store Info

Workspace

Data Source Name

Description

☒ Enabled

Connection Parameters

URL

namespace

☐ create spatial index

charset

☐ memory mapped buffer

Click **Save** button. You will be redirected to the *New Layer chooser* page in order to configure your newly created layer.

Create a Layer

On the *New Layer chooser* page, select your new layer

New Layer chooser

Here is a list of resources contained in the store 'NYC Roads'. Click on the layer you wish to configure

<< < 1 > >>

Results 1 to 1 (out of 1 items)

| Layer with namespace and prefix | Published |
|---------------------------------|---|
| nyc_roads |  |

<< < 1 > >>

Results 1 to 1 (out of 1 items)

The *Edit Layer* page defines the Data and Publishing parameters for a layer. Enter a short *Title* and an *Abstract* for your new layer under *Basic Resource Info*.

nyc_roads:nyc_roads

Configure the resource and publishing information for the current layer

Data **Publishing**

Basic Resource Info

Name
nyc_roads

Title
Subset of NYC roads

Abstract
Shapefile of NYC roads

Generate the layer's bounding boxes by clicking the *Compute* from data and then *Compute from Native bounds*.

Bounding Boxes

Native Bounding Box

| Min X | Min Y | Max X | Max Y |
|-------------|-------------|-------------|------------|
| 984,018.166 | 207,673.095 | 991,906.497 | 219,622.54 |

Compute from data

Lat/Lon Bounding Box

| Min X | Min Y | Max X | Max Y |
|---------|--------|---------|--------|
| -74.001 | 40.737 | -73.972 | 40.769 |

Compute from native bounds

Set the layer's style by switching to the *Publishing* tab. You can select an existing style from the *Default Style* drop down list or create your new own style using the SLD editor. Then click the *Save* button.

Preview the Layer

You can verify that your layer is published properly by navigating to the *Layer Preview* screen.






Layer Preview

List of all layers configured in GeoServer and provides previews in various formats for each.

<< < 1 2 > >>

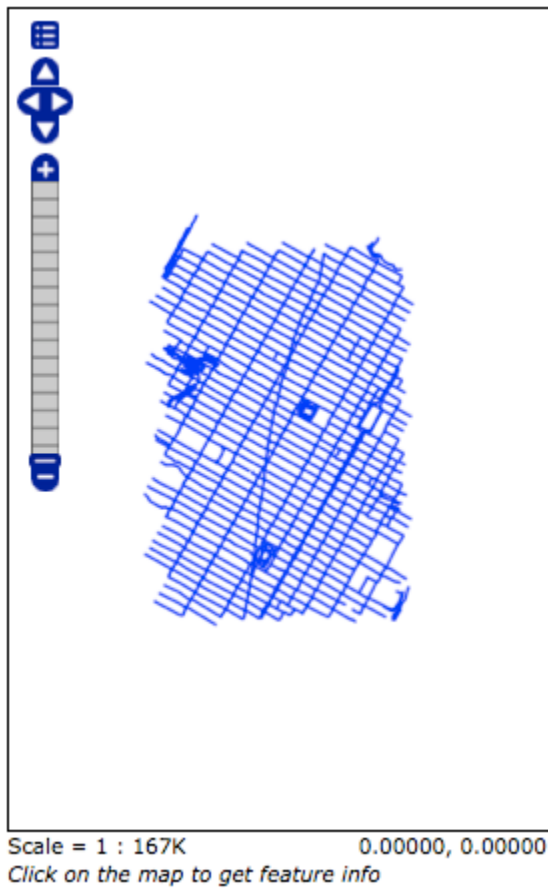
Results 1 to 25 (out of 26 items)

Search

| Type | Name | Title | Common Formats | All Formats |
|---|---------------------|---|--|-------------|
|  | nunc:Arc_Sample | A sample ArcGrid file | OpenLayers KML | Select one |
|  | nunc:Img_Sample | North America sample imagery | OpenLayers KML | Select one |
|  | nunc:Pk50095 | Pk50095 is a A raster file accompanied by a spatial data file | OpenLayers KML | Select one |
|  | nunc:mosaic | Sample PNG mosaic | OpenLayers KML | Select one |
|  | nyc_roads:nyc_roads | Subset of NYC roads | OpenLayers KML GML | Select one |

Click on the *OpenLayers* link in the *Common Formats* column.

An OpenLayers map will load in a new page and should display the Shapefile data with the selected style. You can use the Preview Map to zoom and pan around the dataset, as well as display the attributes of features.



Publishing a WMS service with Mapserver technology

The proposed procedure is derived from information available online at the following location:

http://mapserver.org/ogc/wms_server.html#setting-up-a-wms-server-using-mapserver

Setup a Mapfile for your WMS

Each instance of WMS server that you setup needs to have its own mapfile. In order to create a WMS service you need to configure mandatory metadata fields required to produce a valid GetCapabilities output.

Here is the list of parameters and metadata items that usually optional with MapServer, but are required (or strongly recommended) for a WMS configuration:

At the **MAP** level:

Map NAME

WMS Capabilities requires a Name and a Title tag for every layer. The Map's NAME and wms_title metadata will be used to set the root layer's name and title in the GetCapabilities XML output. The root layer in the WMS context corresponds to the whole mapfile and includes all layers defined in the current map configuration.

Map PROJECTION

WMS servers have to advertise the projection in which they are able to serve data using EPSG projection codes.

```
PROJECTION
    "init=epsg:4326"
END
```

Map METADATA (in the WEB Object):

wms_title

Used in relation with Map NAME element in order to set the proper layer name and title.

wms_srs

This is used to list one or more EPSG codes that the data can be served in.

wms_enable_request

Specify which requests to enable. If not specified, no requests will be enabled. This metadata field is often used to hide layers from the GetCapabilities file. Using a * will enable all types of WMS request.

And for each individual **LAYER**:

Layer NAME

Every individual layer needs its own unique name and title. Layer names are used in *GetMap* and *GetFeatureInfo* requests to refer to layers that should be included in the map output and in the query. Layer names must start with a letter when setting up a WMS server.

Layer PROJECTION

By default layers inherit the SRS of their parent layer (the map's PROJECTION in the MapServer case). For this reason it is not necessary (but still strongly recommended) to provide PROJECTION for every layer. If a layer PROJECTION is not provided then the top-level map projection will be assumed.

Layer METADATA

wms_title

Used in relation with Map NAME element in order to set the proper layer name and title.

Layer STATUS

Layer status is mandatory. Possible values are ON|OFF|DEFAULT

Layer TEMPLATE

You must set the layer TEMPLATE parameter for the layer to be queryable by *GetFeatureInfo* requests. For requests of type "text/html" you should also set the layer HEADER and FOOTER parameters. For other natively supported types, you can control the list of displayed feature attributes by listing them under the *wms_include_items* properties.

Additional information on how to reference geospatial data file or on how to apply cartographic style are available online here:

<http://mapserver.org/mapfile/layer.html>

<http://mapserver.org/mapfile/class.html>

<http://mapserver.org/mapfile/style.html>

Enabling GetFeatureInfo support

In order to enable the support for GetFeatureInfo, you must set the layer TEMPLATE parameter on the layer you want to be queryable. The template refers to a static file used to configure the display of the request output. For HTML format, we recommend enhancing the output with WET4 classes in order to have a nicer display in any GoC web application.

Additional information on Templating are available online here:

<http://mapserver.org/mapfile/template.html>

Test your WMS server

You can validate the content of the service capabilities file by invoking the GetCapabilities URL of your service. You request this document by appending the following query string to the WMS service endpoint "SERVICE=WMS&VERSION=1.1.1&REQUEST=GetCapabilities", for instance:

`http://myhost/cgi-bin/mapserv? SERVICE=WMS&VERSION=1.3.0&REQUEST=GetCapabilities`

You should receive a complete XML capabilities document. You can search for *WARNING* keyword. Usually, MapServer puts warning comments starting with "<!--WARNING: " everywhere a missing metadata element was found.

You can also test the *GetMap* request by invoking your wms service endpoint with mandatory URL parameters. Here is an example of GetMap request:

`http://myhost/cgi-bin/mapserv?map=my_mapfile.map&SERVICE=WMS&VERSION=1.3.0
&REQUEST=GetMap&LAYERS=my_layer&STYLES=&CRS=EPSG:4326
&BBOX=-141.537,40.8775,-49.9603,83.8009&WIDTH=400&HEIGHT=300&FORMAT=image/png`

Sample WMS MapServer Mapfile

MAP

```
NAME "WMS-test"  
STATUS ON  
SIZE 400 300  
EXTENT -2200000 -712631 3072800 3840000  
UNITS METERS  
SHAPEPATH "../data"  
IMAGECOLOR 255 255 255  
FONTSET "../etc/fonts.txt"
```

WEB

```
IMAGEPATH "/tmp/"  
IMAGEURL "/tmp/"
```

```

METADATA
    "wms_title"          "WMS Demo Server"  ##required
    "wms_srs"            "EPSG:42304 EPSG:42101 EPSG:4269 EPSG:4326"
    "wms_enable_request" "*"               ##necessary
END
END # Web

PROJECTION
    "init=epsg:3978"      ##required
END

SYMBOL
    NAME "circle"
    TYPE ellipse
    POINTS 1 1 END
END # Symbol

#
# Start of layer definitions
#

LAYER
    NAME "park"
    METADATA
        "wms_title"      "Parks"      ##required
    END
    TYPE POLYGON
    STATUS OFF
    DATA park
    PROJECTION
        "init=epsg: 3978"    ##recommended
    END
    CLASS
        NAME "Parks"
        STYLE
            COLOR 200 255 0
            OUTLINECOLOR 120 120 120
        END # Style
    END # Class
END # Layer

LAYER
    NAME popplace
    METADATA
        "wms_title"      "Cities"      ##required

```



```

END
TYPE POINT
STATUS ON
DATA popplace
PROJECTION
    "init=epsg:3978"    ##recommended
END
CLASS
    NAME "Cities"
    STYLE
        SYMBOL "circle"
        SIZE 8
        COLOR 0 0 0
    END # Style
END # Class
END # Layer

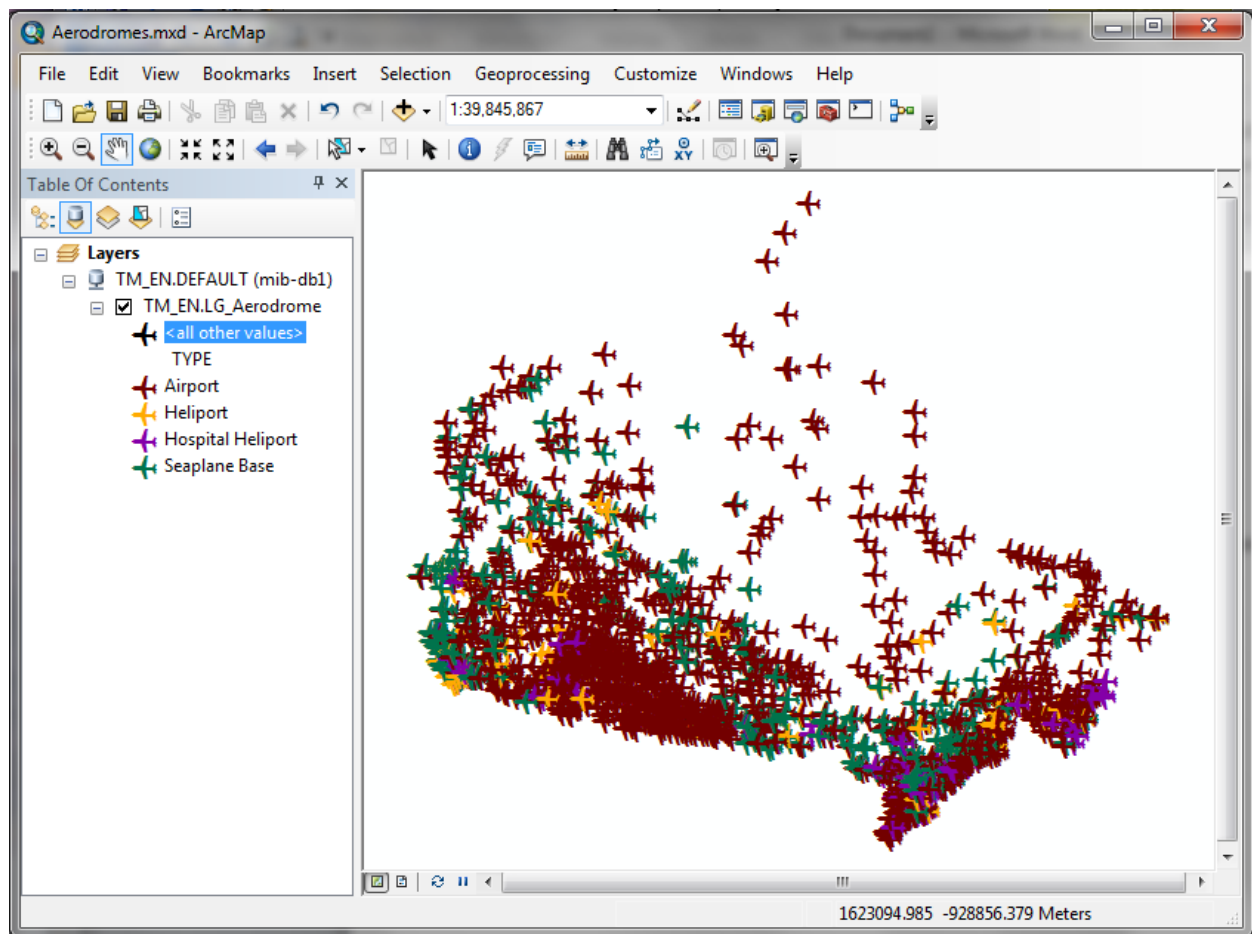
END # Map File

```

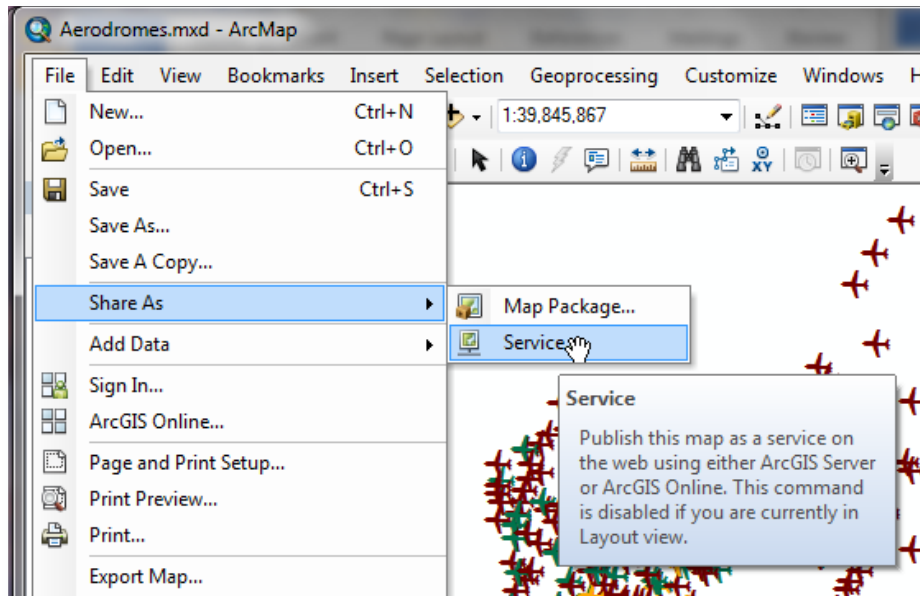
Publishing a mapping service with Esri technology

For an in-depth look about publishing services with ESRI technology, you can visit their web site at http://resources.arcgis.com/en/help/main/10.1/index.html#/How_to_publish_a_service/0154000004n3000000/. Here, we will describe the general process.

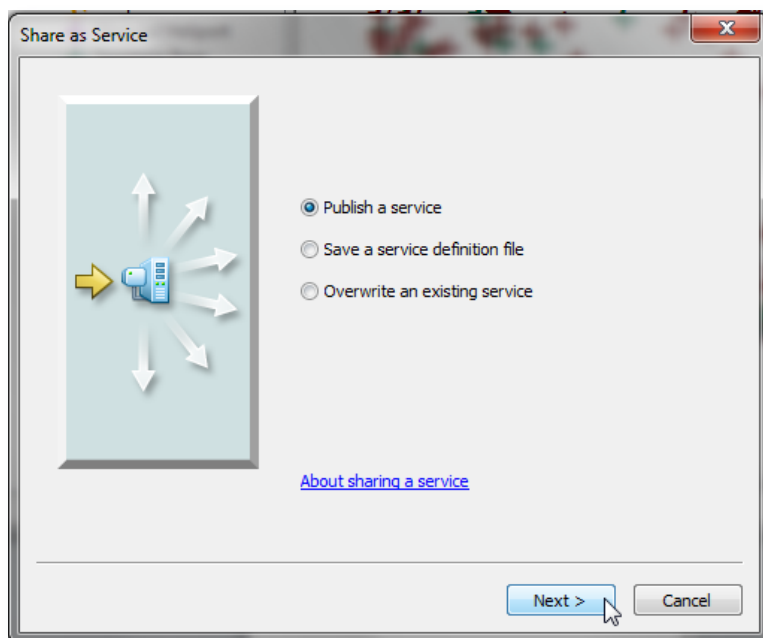
1. Create an MXD file using ArcGIS Desktop. Here, we created a simple map showing aerodromes in Canada:
2. Make sure to use one of the following symbol rendering :
 - Simple Rendering (all data in layer utilizes 1 symbol)
 - Unique Value Rendering (symbol utilized is defined by the value of one or more attributes)
 - Class Break Rendering (symbol is defined by the magnitude of certain values i.e. population, emission quantity, etc.)



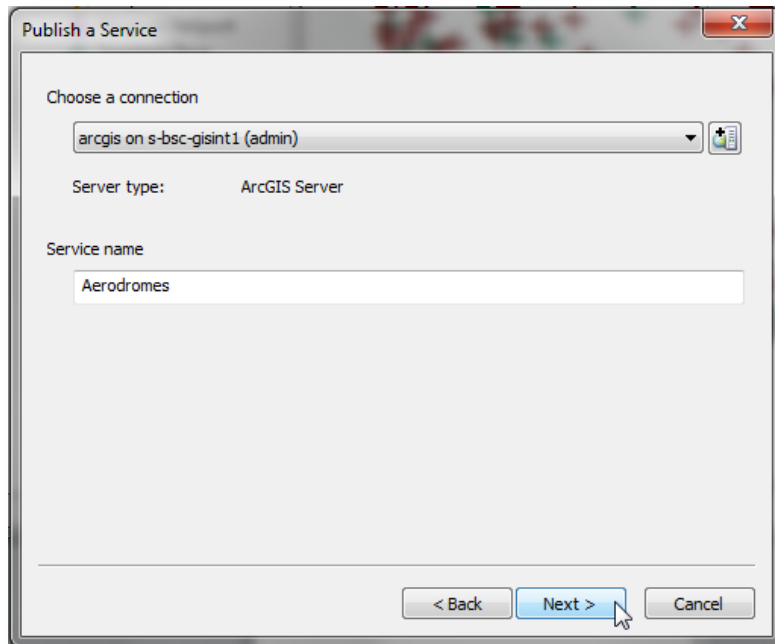
3. Choose the attribute to display but make sure the OBJECTID and the SHAP attributes are not hidden
4. To publish this map as a service, the ArcGIS Administrator must have given you privileges to publish on the server.
5. Start the publishing process by selecting "Service" under "Share As":



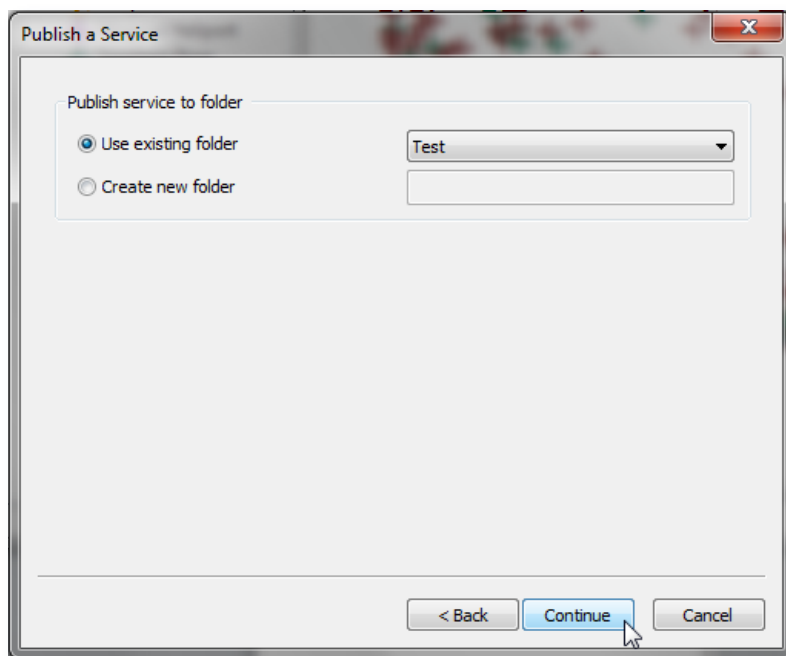
6. This opens a wizard to help publish you map. Select “Publish a service” and hit “Next”.



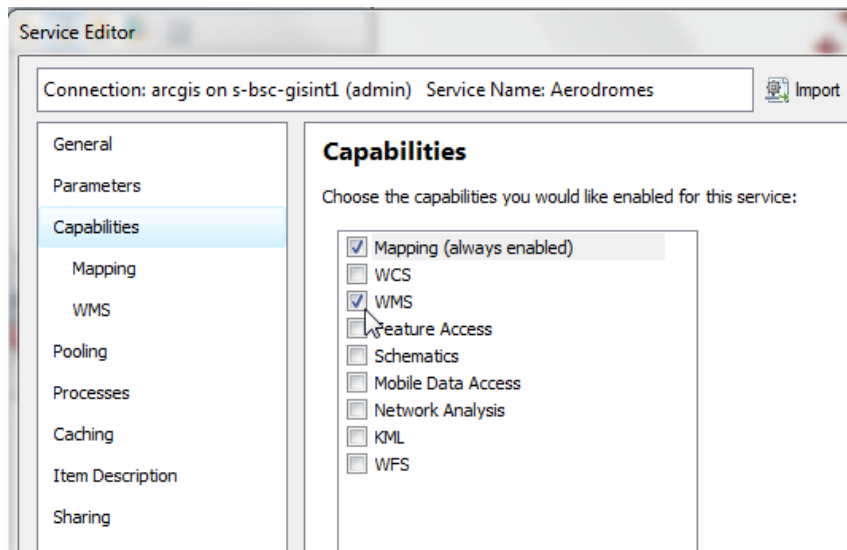
7. Select the server on which you want to publish and type in your desired service name, then click “Next”



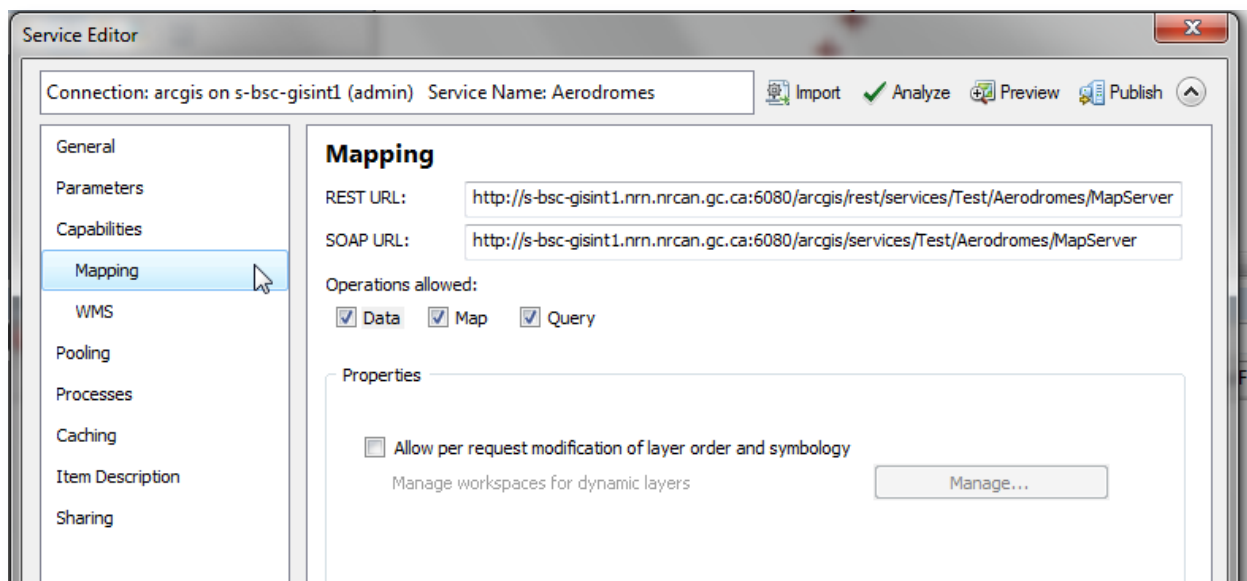
8. Select the folder in which you want to publish your service, or create a new one, and click “Continue”



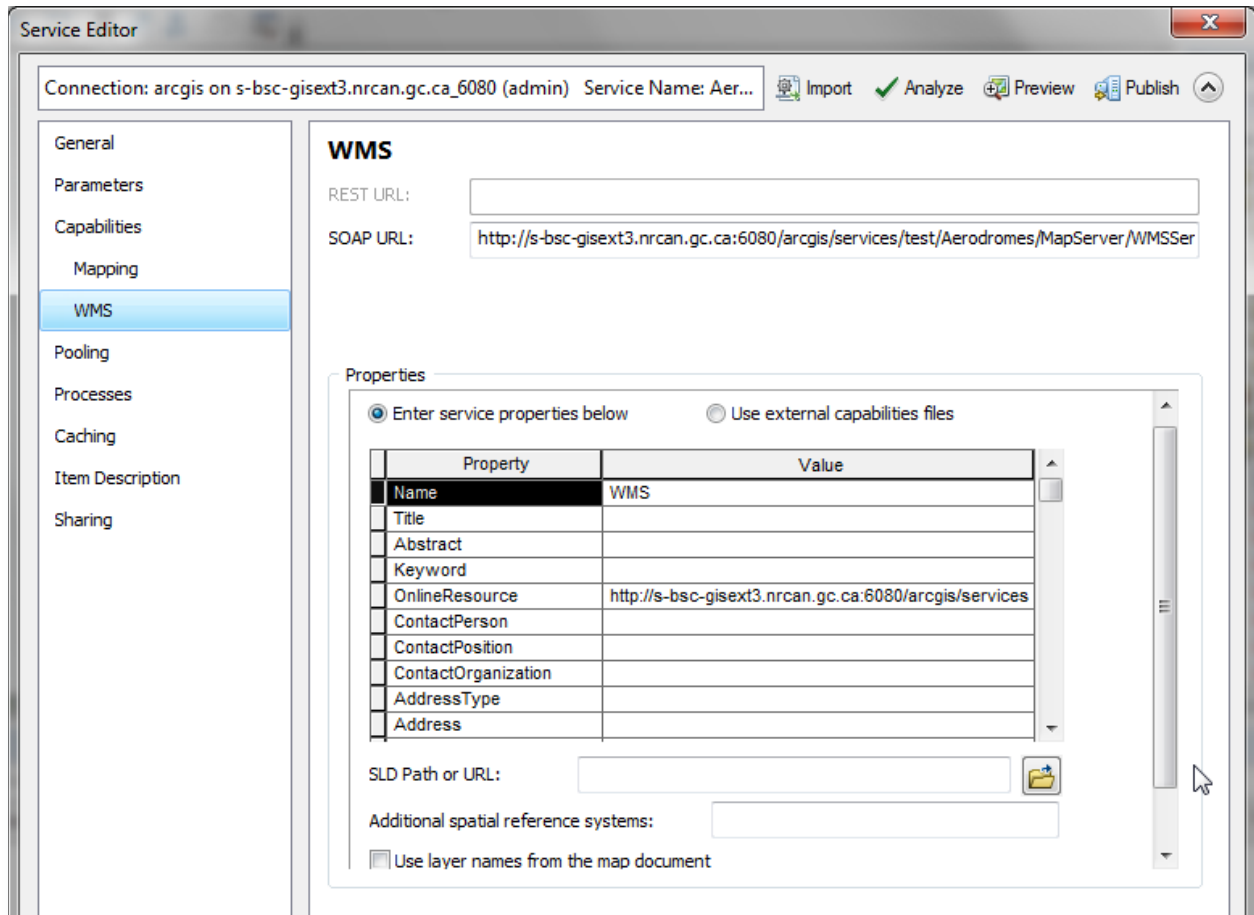
In these next steps, we have to setup the service capabilities. We will only cover items in the “Capabilities” section. This is where you tell ArcGIS Server the type of services you want to support. By default, you get the “Mapping” service. Here you can add several other types of services, like OGC services. Let’s add WMS.


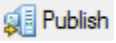


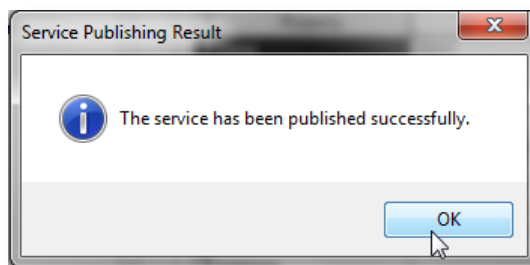
9. Next, we can edit the capabilities of each service type. Click the “Mapping” capability. In here we can control the different options for the mapping service. We will use the defaults.
10. Make sure the DATA operation is enabled



11. Click the “WMS” capability. Here, you can edit the properties of the WMS service, or select to use external capabilities files. Make sure you use Layer names from Map Document.

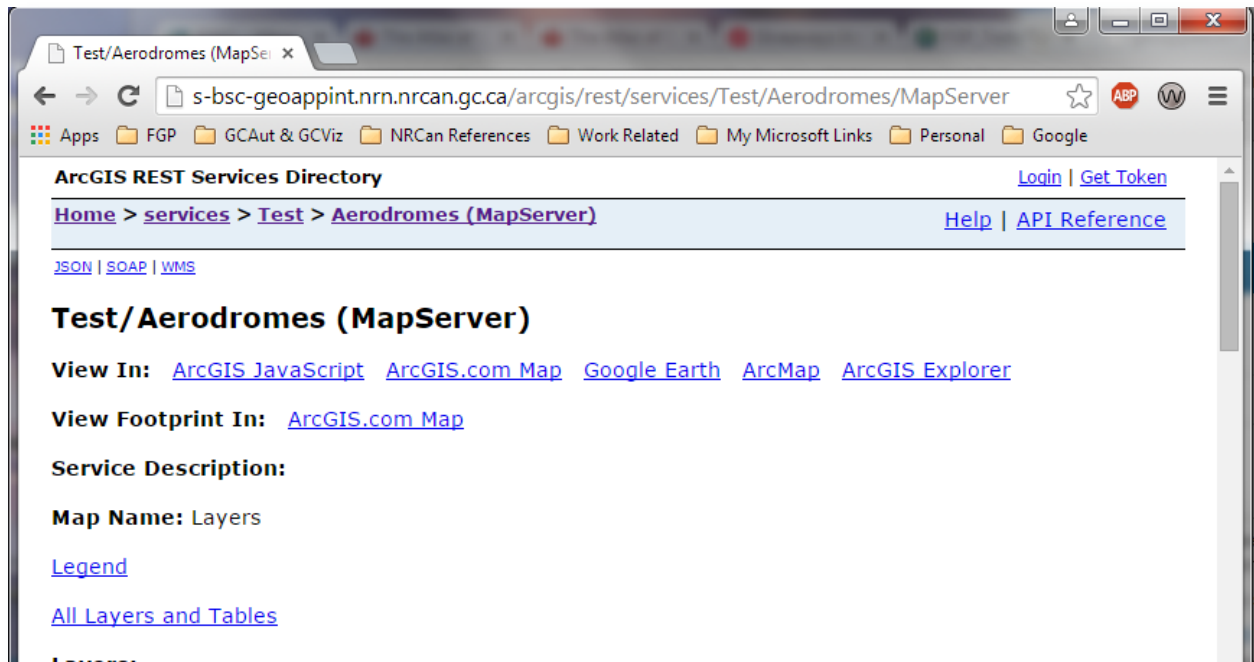


12. Now that the service capabilities have been defined, we would press the  button to analyze our service before it is published. This will create a report for us outlining any problems we would need to fix before we publish the service. After correcting any problems, we would come back to this step and press the  button to publish our service.
13. You will get notified when the service has been published.

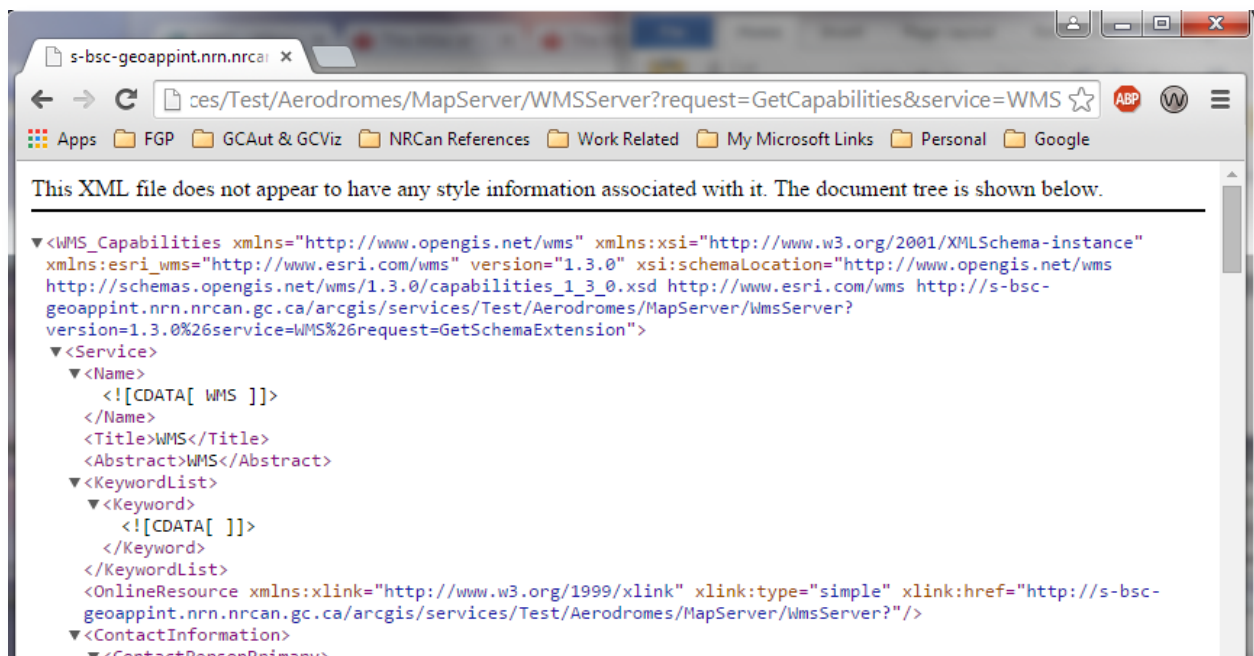


14. Now, we want to use the service we just created. To use the "Mapping" service, we use the REST endpoint. If we look at it in a browser, we can see that it provides links for us to view our

service in different viewers such as a custom JavaScript application, Google Earth and ArcMap to name a few.



On the page, you can also see a link named “WMS”. This is our link to the WMS service. By clicking this link, we get the WMS capabilities file for the service.



Server Considerations:

To style the output of the service, you will need to replace the content of the file “featureinfo_text_html.xml” in the folder “C:\Program Files\ArcGIS\Server\Styles\WMS” with something like the following, and this must be done on all the servers of your ArcGIS Server cluster.



featureinfo_text_html.xml

Also, in the RAMP configuration file, you will need to replace

```
"featureInfo":{"mimeType":"text/plain"
```

by

```
"featureInfo":{"mimeType":"text/html"
```

for each WMS layer. If you want a more customized output, you will need to create a specific style sheet for each layer, add a parameter in the configuration file for the WMS services, build the necessary style files and install them on the server(s). You would also have to modify the JavaScript code to use your new parameter.

ⁱ <http://stackoverflow.com/questions/226108/what-is-a-web-service-in-plain-english> (accessed 2105-02-05)

ⁱⁱ <http://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/#webservice> (accessed 2015-02-05)

ⁱⁱⁱ <https://www.w3.org/2002/ws/>

^{iv} <http://stackoverflow.com/questions/226108/what-is-a-web-service-in-plain-english> (accessed 2015-02-05)

^v <http://www.opengeospatial.org/ogc> (accessed 2015-02-05)

^{vi} <http://www.opengeospatial.org/ogc> (accessed 2015-02-05)

^{vii} <http://www.techopedia.com/definition/12888/location-based-services-lbs> (accessed 2015-02-10)

^{viii} Geographic Standards Guide, GeoConnections, January 2015 (unpublished draft)

^{ix} <http://resources.arcgis.com/en/help/main/10.1/index.html#/015400000383000000> (accessed 2015-02-10)

^x <http://www.opengeospatial.org/standards/wms> (accessed 2015-02-05)

^{xi} SLD (Styled Layer Description) is a OGC standard providing rule based instructions to render map features based on their attributes. It can be used to “query” the data store by providing rendering rules just for feature matching query parameters (the result is still an image) and not rendering the others. SLD support is not required for WMS 1.3.0 (ISO 19128) because SLD is not a ISO standard.

^{xii} <http://www.nrcan.gc.ca/earth-sciences/geomatics/canadas-spatial-data-infrastructure/standards-policies/8940#Additional%20Information> (accessed 2015-02-05)

^{xiii} <http://www.ogcnetwork.net/wcps> (accessed 2015-02-10)