

OGC Testbed-13
GeoPackage Engineering Report

Table of Contents

1. Summary	4
1.1. Requirements	4
1.2. Key Findings and Prior-After Comparison	4
1.2.1. Prior Work	4
1.3. What does this ER mean for the Working Group and OGC in general	5
1.4. Document contributor contact points	5
1.5. Future Work	5
1.6. Foreword	6
2. References	7
3. Terms and definitions	8
3.1. asynchronous	8
3.2. portrayal	8
3.3. GeoPackage	8
3.4. GeoPackage extension	8
3.5. styling	8
3.6. symbology	8
3.7. topographic data	9
4. Abbreviated terms	10
5. Overview	11
5.1. clause-requirements	11
6. Solutions	12
6.1. Web Processing Service (WPS) Solution	12
6.1.1. Support for SE Style Rendering	12
6.2. Client Solutions	16
6.2.1. Support for Layer Ordering	16
6.2.2. Support for Universal Transverse Mercator (UTM) tiles	17
6.2.3. Support for Dynamic Styling	17
Appendix A: GeoPackage WPS Implementation	19
A.1. Global Properties	19
A.2. WMS Offerings Properties	20
A.3. WMTS Offerings Properties	21
A.4. FileGDB Offerings Properties	22
A.5. WPS Operations	23
A.5.1. Execute (Context Document)	23
A.5.2. GetStatusRequest (JobId)	28
A.5.3. StatusInfo Response (JobId)	28
A.5.4. GetResult Request (JobId)	29
Appendix B: Static Feature Symbology Extension (Normative)	30

B.1. Extension Title	30
B.2. Introduction	30
B.3. Extension Author	30
B.4. Extension Template	30
B.5. Extension Type	31
B.6. Applicability	31
B.7. Scope	31
B.8. Requirements	31
B.8.1. Table Definitions	31
Appendix C: Dynamic Feature Symbology Extension (Normative)	38
C.1. Extension Author	38
C.2. Extension Name or Template	38
C.3. Extension Type	38
C.4. Applicability	39
C.5. Scope	39
C.6. Requirements	39
C.6.1. Table Definitions	39
Appendix D: UTM Tile Matrices	43
D.1. Calculating a usable extent	44
D.2. A UTM Example	45
Appendix E: Revision History	48
Appendix F: Bibliography	49

Publication Date: 2018-01-26

Approval Date: 2018-01-24

Posted Date: 2017-11-15

Reference number of this document: OGC 17-027

Reference URL for this document: <http://www.opengis.net/doc/PER/t13-UG001>

Category: Public Engineering Report

Editor: Robert Cass

Title: OGC Testbed-13: GeoPackage Engineering Report

OGC Engineering Report

COPYRIGHT

Copyright © 2018 Open Geospatial Consortium. To obtain additional rights of use, visit <http://www.opengeospatial.org/>

WARNING

This document is not an OGC Standard. This document is an OGC Public Engineering Report created as a deliverable in an OGC Interoperability Initiative and is not an official position of the OGC membership. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard. Further, any OGC Engineering Report should not be referenced as required or mandatory technology in procurements. However, the discussions in this document could very well lead to the definition of an OGC Standard.

LICENSE AGREEMENT

Permission is hereby granted by the Open Geospatial Consortium, ("Licensor"), free of charge and subject to the terms set forth below, to any person obtaining a copy of this Intellectual Property and any associated documentation, to deal in the Intellectual Property without restriction (except as set forth below), including without limitation the rights to implement, use, copy, modify, merge, publish, distribute, and/or sublicense copies of the Intellectual Property, and to permit persons to whom the Intellectual Property is furnished to do so, provided that all copyright notices on the intellectual property are retained intact and that each person to whom the Intellectual Property is furnished agrees to the terms of this Agreement.

If you modify the Intellectual Property, all copies of the modified Intellectual Property must include, in addition to the above copyright notice, a notice that the Intellectual Property includes modifications that have not been approved or adopted by LICENSOR.

THIS LICENSE IS A COPYRIGHT LICENSE ONLY, AND DOES NOT CONVEY ANY RIGHTS UNDER ANY PATENTS THAT MAY BE IN FORCE ANYWHERE IN THE WORLD. THE INTELLECTUAL PROPERTY IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE DO NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE INTELLECTUAL PROPERTY WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE INTELLECTUAL PROPERTY WILL BE UNINTERRUPTED OR ERROR FREE. ANY USE OF THE INTELLECTUAL PROPERTY SHALL BE MADE ENTIRELY AT THE USER'S OWN RISK. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR ANY CONTRIBUTOR OF INTELLECTUAL PROPERTY RIGHTS TO THE INTELLECTUAL PROPERTY BE LIABLE FOR ANY CLAIM, OR ANY DIRECT, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM ANY ALLEGED INFRINGEMENT OR ANY LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR UNDER ANY OTHER LEGAL THEORY, ARISING OUT OF OR IN CONNECTION WITH THE IMPLEMENTATION, USE, COMMERCIALIZATION OR PERFORMANCE OF THIS INTELLECTUAL PROPERTY.

This license is effective until terminated. You may terminate it at any time by destroying the Intellectual Property together with all copies in any form. The license will also terminate if you fail to comply with any term or condition of this Agreement. Except as provided in the following sentence, no such termination of this license shall require the termination of any third party end-user sublicense to the Intellectual Property which is in force as of the date of notice of such termination. In addition, should the Intellectual Property, or the operation of the Intellectual Property, infringe, or in LICENSOR's sole opinion be likely to infringe, any patent, copyright, trademark or other right of a third party, you agree that LICENSOR, in its sole discretion, may terminate this license without any compensation or liability to you, your licensees or any other party. You agree upon termination of any kind to destroy or cause to be destroyed the Intellectual Property together with all copies in any form, whether held by you or by any third party.

Except as contained in this notice, the name of LICENSOR or of any other holder of a copyright in all or part of the Intellectual Property shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Intellectual Property without prior written authorization of LICENSOR or such copyright holder. LICENSOR is and shall at all times be the sole entity that may authorize you or any third party to use certification marks, trademarks or other special designations to

indicate compliance with any LICENSOR standards or specifications.

This Agreement is governed by the laws of the Commonwealth of Massachusetts. The application to this Agreement of the United Nations Convention on Contracts for the International Sale of Goods is hereby expressly excluded. In the event any provision of this Agreement shall be deemed unenforceable, void or invalid, such provision shall be modified so as to make it valid and enforceable, and as so modified the entire Agreement shall remain in full force and effect. No decision, action or inaction by LICENSOR shall be construed to be a waiver of any rights or remedies available to it.

None of the Intellectual Property or underlying information or technology may be downloaded or otherwise exported or reexported in violation of U.S. export laws and regulations. In addition, you are responsible for complying with any local laws in your jurisdiction which may impact your right to import, export or use the Intellectual Property, and you represent that you have complied with any regulations or registration procedures required by applicable law to make this license enforceable.

Chapter 1. Summary

This Engineering Report details the processes and results related to generating GeoPackages developed to contain topographic vector features and supporting symbologies based on The National Map (TNM) product of the United States Geological Survey (USGS).

1.1. Requirements

The Engineering Report shall describe all testbed activities on the integration of USGS Topo Combined Vector Product data to GeoPackage, all experiences made during implementation, including recommendations to the sponsor, and provide any resulting standards change requests to the appropriate standards working groups. The report will also cover these specific items:

- Problems / obstacles encountered working on the USGS specific GeoPackage and geospatial/non geospatial metadata integration requirements.
- Documented process used in meeting the requirements including process for downloading the GeoPackage (encoded with symbology)
- Recommendations for further work needed specific to these Testbed-13 requirements.

The work covered by this Engineering Report falls under the two Testbed-13 deliverables :

- UG102 GeoPackage
- AB102 GeoPackage Mobile client for Symbology and Styles

1.2. Key Findings and Prior-After Comparison

Standardized Universal Transverse Mercator (UTM) tile grids are a worthwhile goal for the OGC. Since work on the testbed began, a set of universal tilesets has been proposed. These tilesets are under consideration for adoption for use in GeoPackage to support map projections other than Geographic and Web Mercator. The adoption of these standardized tilesets will aid interoperability and put the necessary references in the hands of GeoPackage producers to generate GeoPackaged tilesets using Map Projections required by client communities.

OGC Symbology Encoding (SE) has been in existence since 2007 and has some uptake, however, there are some key barriers to its adoption. These barriers are the complexity of a full implementation which requires adherence to ideas such as filters, arbitrary sets of functions, which are beneficial, but require more investment than an engine that draws only the symbology. This testbed used the symbol drawing functionality proposed by SE as a sort of SE "lite". The approach is discussed further in [\[\[annex-gpkg\]\]](#). This approach was reasonable, and the limitations on the symbology were typical shortcomings of SE related to placement of symbols or labels, for which there are no specific directives available in SE.

1.2.1. Prior Work

In OGC testbed-12 similar work stored the same topographic and reference data, using a static proprietary symbology encoding. The Testbed-12 Engineering Report B004-GeoPackage-US-

Topo[[TB12GPKG](#)] documented the work. The GeoPackage Community expressed a need to have well-known standards for rendering geographic features on a map using instructions stored in an interoperable manner. The Testbed-12 approach had limitations which prompted the Testbed-13 work :

- Styling was static and would not change when features were updated
- Symbology in a proprietary format
- Reference data was not projected into local projections such as UTM
- Layer grouping and layer ordering did not exist
- Compliant Metadata was not supported in the GeoPackage

A group has proposed using OGC Web Services (OWS) Context in GeoPackage, which includes support for layer ordering and symbolization to support these requirements. The context encoding has a broad function which does not make use of the table structure and query functionality associated with a GeoPackage.

No definitive uniform Tile Matrix Sets for UTM exist. Work performed within the United States National System for Geospatial Intelligence (NSG) has defined approaches for global Mercator projections[[SIG0014](#)], some of this work may be valuable to understand how to define these tile sets.

1.3. What does this ER mean for the Working Group and OGC in general

This ER should help advance the understanding of GeoPackage, suggest a method of symbology encoding, methods of storing tiles in non global projections. This ER should provide insight into using mobile clients to render GeoPackages in local projections e.g. UTM.

1.4. Document contributor contact points

All questions regarding this document should be directed to the editor or the contributors:

Table 1. Contacts

Name	Organization
Rob Cass	Compusult Ltd.
Kristin Fishburn	USGS

1.5. Future Work

Future work related to the efforts described in this engineering report should focus on:

- the continued support and development of symbology standards within the OGC meet the needs of mapping organizations such as the USGS,
- the extension of the GeoPackage standard to support the encoding of styling information

accompanying feature data stored in a GeoPackage,

- the adoption of common raster tilesets for well-known projections to support easy exchange of tiled raster data through Web Map Tile Service (WMTS) and GeoPackage,
- the extension of OWS Context (OWC) to support well known vector storage formats such as ESRI FileGDB and GeoPackage,
- the extension of OWC to support referencing and querying "layers" of information contained in traditionally opaque formats such as GeoPackage, KML, FileGDB; and
- a standard for internally referencing GeoPackage entities and attributes that use querying methods.

1.6. Foreword

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

Chapter 2. References

- [OGC 05-077r4, OGC® Symbology Encoding Implementation Specification](http://portal.opengeospatial.org/files/?artifact_id=16700) [http://portal.opengeospatial.org/files/?artifact_id=16700]
- [OGC 06-042, OGC® Web Map Service \(WMS\) Implementation Specification](http://portal.opengeospatial.org/files/?artifact_id=14416) [http://portal.opengeospatial.org/files/?artifact_id=14416]
- [OGC 07-057r7, OGC® Web Map Tile Service Implementation Standard](http://portal.opengeospatial.org/files/?artifact_id=35326) [http://portal.opengeospatial.org/files/?artifact_id=35326]
- [OGC 14-065, OGC® WPS 2.0 Interface Standard](http://docs.opengeospatial.org/is/14-065/14-065.html) [http://docs.opengeospatial.org/is/14-065/14-065.html]
- [OGC Geopackage Encoding Standard](http://www.geopackage.org/spec/) [http://www.geopackage.org/spec/]
- [TrueType 1.0 Specification](https://www.microsoft.com/en-us/Typography/SpecificationsOverview.aspx) [https://www.microsoft.com/en-us/Typography/SpecificationsOverview.aspx]
- [Testbed-12 GeoPackage US Topo Engineering Report](http://docs.opengeospatial.org/per/16-037.html) [http://docs.opengeospatial.org/per/16-037.html]
- Map Projections for Tiled Raster Graphics, NGA.SIG.0014_1.0_PROJRAS Version 1.0

Chapter 3. Terms and definitions

For the purposes of this report, the definitions specified in Clause 4 of the OWS Common Implementation Standard [OGC 06-121r9](https://portal.opengeospatial.org/files/?artifact_id=38867&version=2) [https://portal.opengeospatial.org/files/?artifact_id=38867&version=2] shall apply. In addition, the following terms and definitions apply.

3.1. asynchronous

A form of client-server communication in which the server responds immediately that a requested operation has begun. A client can make requests to the server for the status and result of the operation, or the client can receive notification of the status and result. This form of communication is typically used when the operation takes a significant amount of time.

3.2. portrayal

The process of using symbology to draw map features.

3.3. GeoPackage

An sqlite database that stores vector features, raster imagery, non-spatial data and extended data using a standard defined at <http://www.geopackage.org/>.

3.4. GeoPackage extension

An additional information encoding standard used in a GeoPackage defined using the GeoPackage extension mechanism.

3.5. styling

A portrayal sub-process that uses a feature's attributes and display context to determine the symbology used to draw that feature.

3.6. symbology

A system of symbols used to draw map features.

3.7. topographic data

Data used to represent man-made and natural features as well as contour intervals. This data is made available in digital form to support portrayal by computers on-screen and on paper.

Chapter 4. Abbreviated terms

- API Application Program Interface
- GPKG GeoPackage
- JSON JavaScript Object Notation
- OWC OWS Context Document
- SE OGC Symbology Encoding
- SLD OGC Styled Layer Descriptor
- TIFF Tagged Interchange File Format
- TTF TrueType Font
- UTM Universal Transverse Mercator
- WMS Web Map Service
- WMTS Web Map Tile Service
- WPS OGC Web Processing Service
- XML eXtensible Markup Language

Chapter 5. Overview

This Engineering Report records the work done for Testbed-13 to support the generation of USGS topographic data GeoPackages so that they can be visualized in a client according to USGS requirements.

5.1. clause-requirements

The Engineering Report shall describe all testbed activities on the integration of USGS Topo Combined Vector Product data to GeoPackage, all experiences made during implementation, including recommendations to the sponsor, and provide any resulting standards change requests to the appropriate standards working groups. The report will also cover these specific items:

- Problems / obstacles encountered working on the USGS specific GeoPackage and geospatial/non geospatial metadata integration requirements.
- Documented process used in meeting the requirements including process for downloading the GeoPackage (encoded with symbology)
- Recommendations for further work needed specific to these Testbed-13 requirements.

The work covered by this Engineering Report falls under the two Testbed-13 deliverables :

- UG102 GeoPackage
- AB102 GeoPackage Mobile client for Symbology and Styles

Chapter 6. Solutions

6.1. Web Processing Service (WPS) Solution

To generate useful GeoPackages by aggregating distributed data sources, such as Web Map Tile Service (WMTS), Web Map Service (WMS), and network available FileGDBs and style sheets, a WPS handles the registration of these resources. Clients invoke this WPS to create GeoPackages.

The WPS takes Extensible Markup Language (XML) requests to :

- Create, retrieve, GeoPackages
- Monitor the status of the GeoPackage Creation Process
- Identify the URL of the resulting GeoPackage to download

The sequence diagram below illustrates the process used for generating a GPKG from USGS resources.

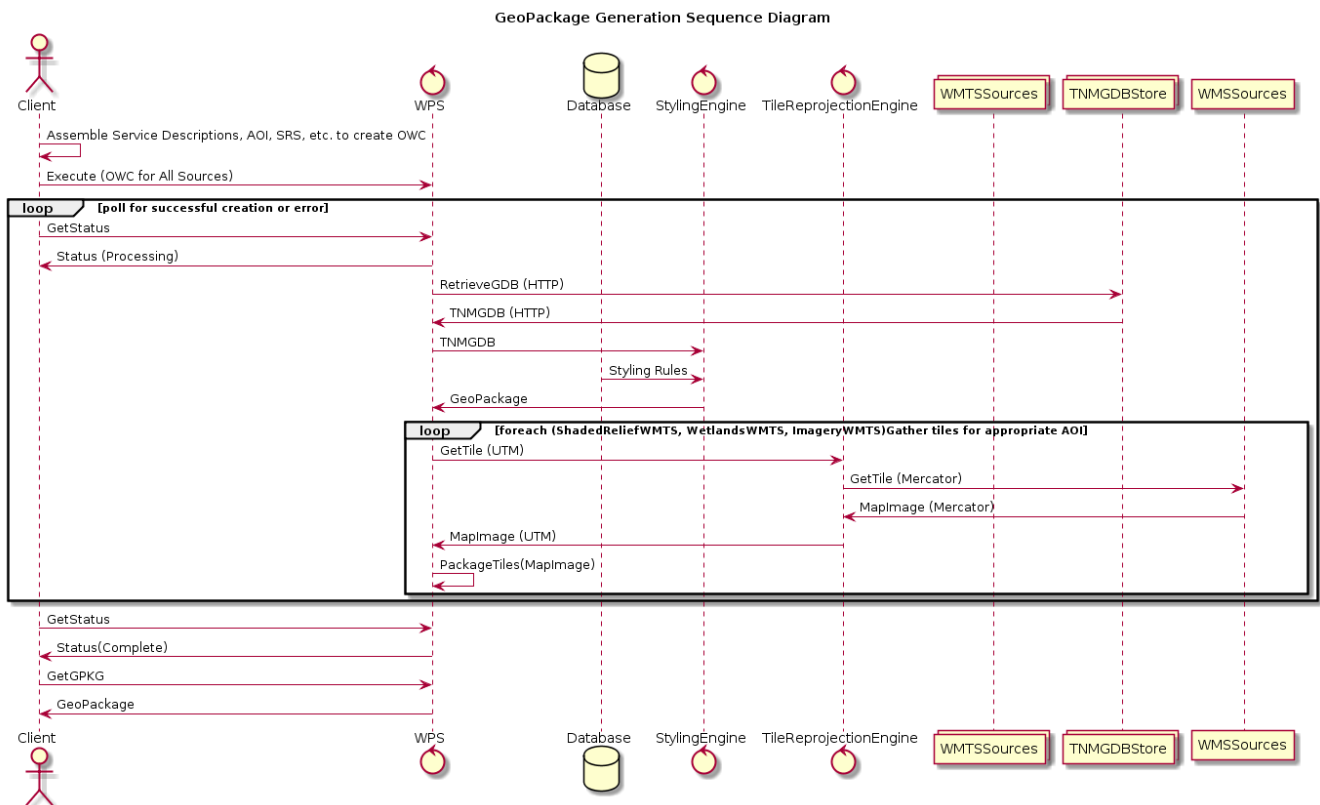


Figure 1. GeoPackage Generation Sequence Diagram

Each operation used in the general flow follows the WPS 2.0 specification[[WPS](#)]

[WPS Operations](#) describes the WPS operations illustrated in [Figure 1](#) in detail.

6.1.1. Support for SE Style Rendering

Testbed-13 provided an opportunity to test the effectiveness of the OGC Symbology Encoding (SE) standard as the encoding of choice used in a GeoPackage styling extension. This extension was used in Testbed-12 to style topographic vector feature data provided by the USGS.

In order to maintain a reasonable scope during the testbed a subset of SE functionality was implemented. The graphic rendering elements were kept, but other more complicated elements were discarded. This approach reduced scope, and avoided duplication of styling filters. To have the filters expressed in SE, when the filter rules were also implemented in a GeoPackage extension was unnecessary.

The diagrams shown immediately below illustrate the subset of SE symbolizers used to render the USGS topographic data.

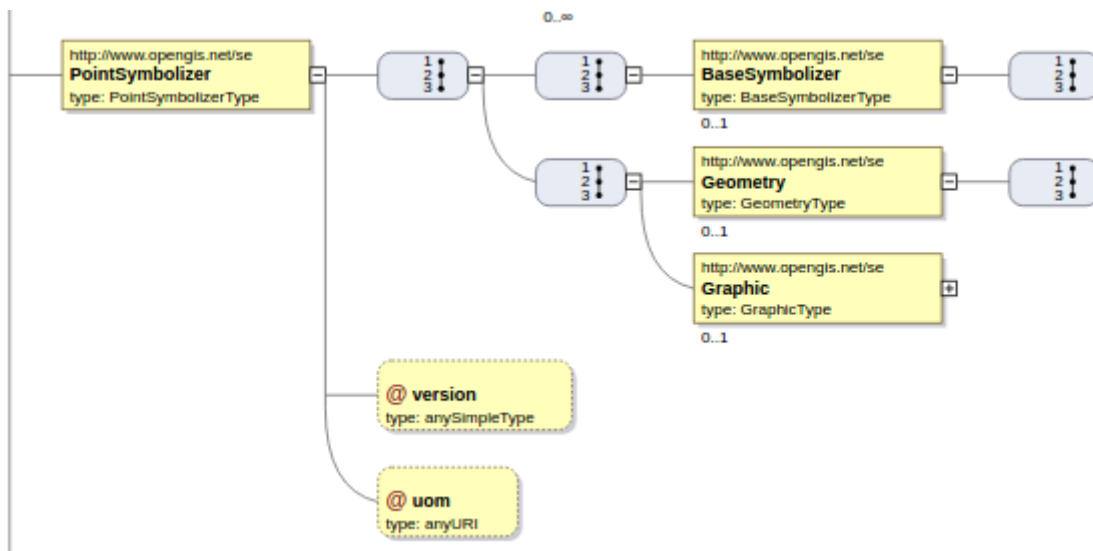


Figure 2. Point Symbolizer

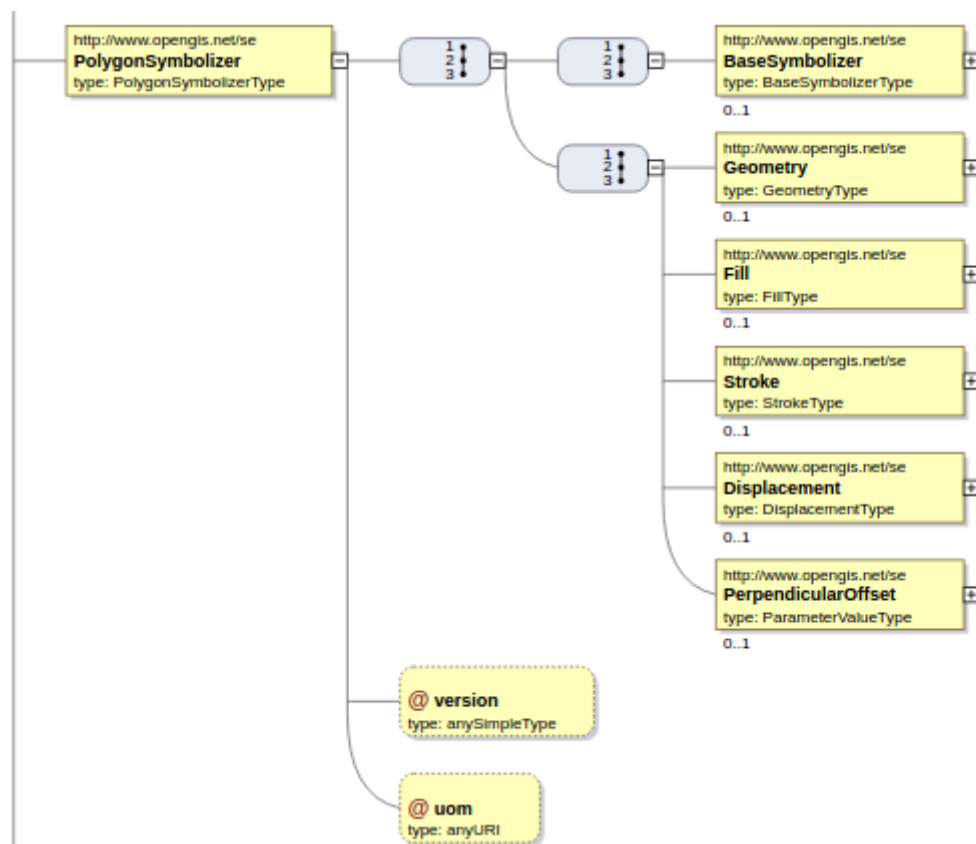


Figure 3. PolygonSymbolizer

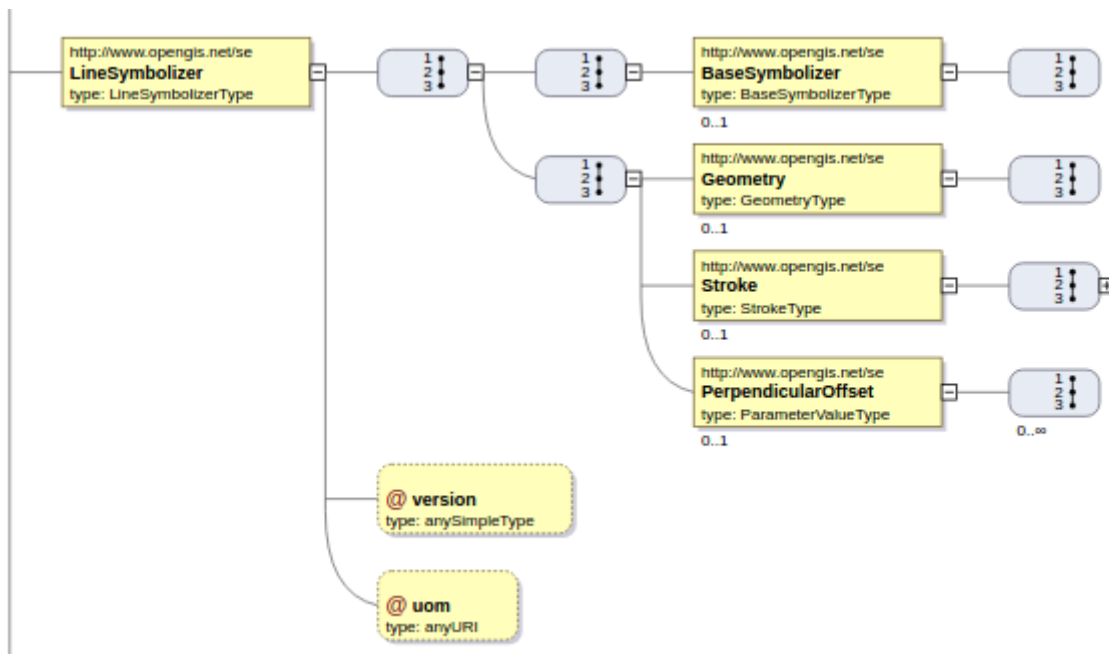


Figure 4. LineSymbolizer

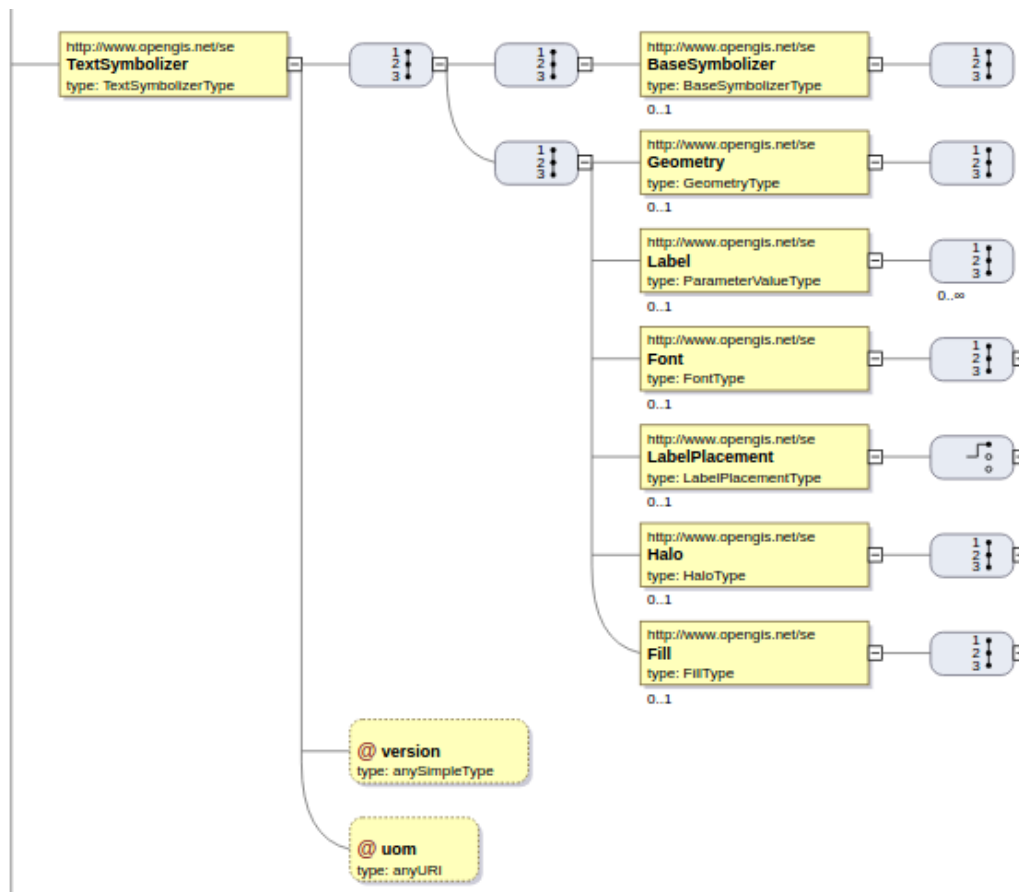


Figure 5. TextSymbolizer

This approach was successful. The following issues were found when trying to implement styling rules in SE :

- Inline content or URLs are used for images in SE. In the styling extension, images are stored in a separate table. There was a need to refer to the images stored in the GeoPackage. This issue is not a fault of SE, but illustrates a common need for this form of referencing in a GeoPackage. An "extension" of SE was encoded using sql to define the database query to extract the image. It has

since been pointed out that this approach is dangerous as sql could be introduced to do something other than query the images table. This danger highlights the need for a standardized method for internal gpkg references that is approved and reviewed.

- There is no real support for shield symbology such as highway shields which require a 2 pass approach [3]. In order to support this with the current implementation, 2 layers would be required to support the drawing of the shield symbol, then another layer to draw the shield label on top. In order to get around this, parameters were introduced for line text symbolizers so that they would be drawn at the vertices of the line feature. The shield shaped ticks for the line symbolizers are also drawn on the vertex of the line feature. Having both of these drawn in the correct order draws something like a shield. Both Bocher[3] and the GeoServer team recognized this shortcoming and either requested it or implemented the functionality as an extension. This requirement should be reviewed and supported in future versions of SE.

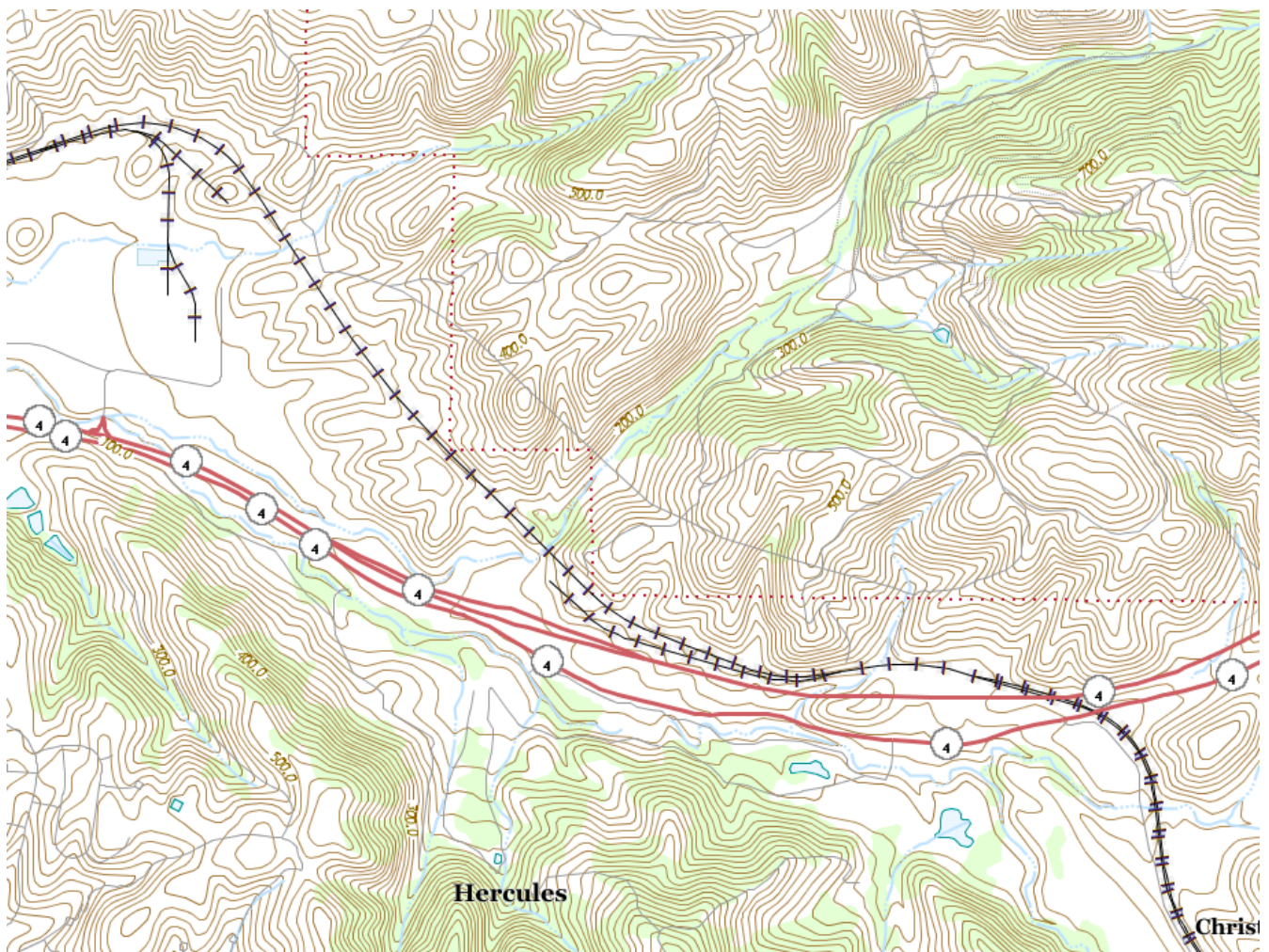


Figure 6. Shield Symbolizers

Embedded font support was also added to the GeoPackage styling extension. The font is stored in a gpkgext_custom_fonts table. The binary format is intended to be Truetype format. Client systems would need to extract the font from the table and register the font in the supporting operating system. SE Text Symbolizers using these fonts would refer to them by the registered typeface name.

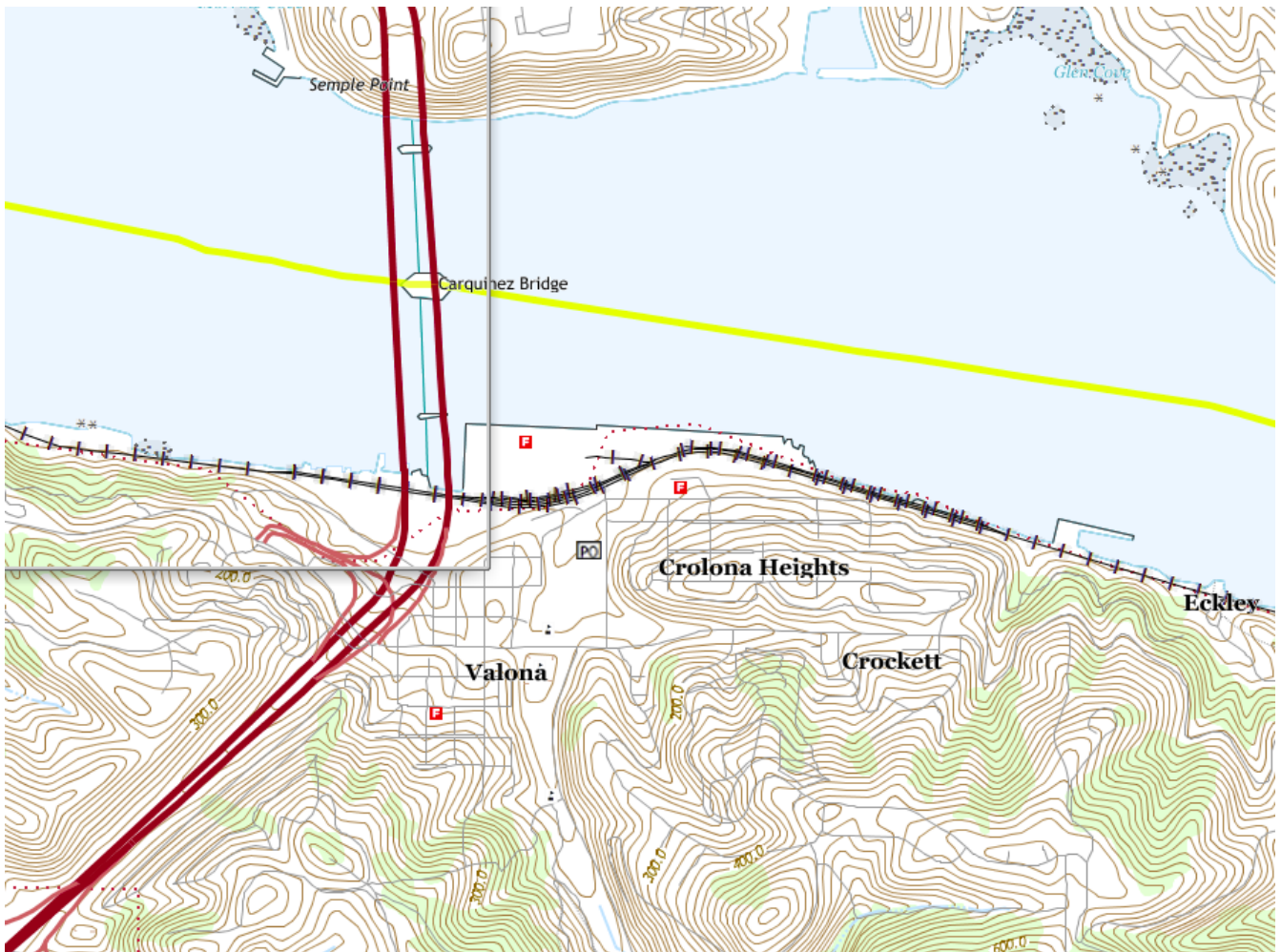


Figure 7. Font Support

The outcome for this experiment was a success not only for SE, but also for the styling extension. The stored format of the style did not affect the data model of the extension. Additionally, there appeared to be no significant impact from storing the styles in SE.

6.2. Client Solutions

Supporting the newly created GeoPackages required few changes in the client. Any display client will translate the textual style encoding to the internal styling system. Whether the style is encoded in JavaScript Object Notation (JSON) or SE is irrelevant once the system translates the data and stores it in an efficient manner.

As part of Testbed-13, software clients were modified to support the translation of the styling data in SE to a lower level form used to do drawing. These clients were also enhanced to use the extended font support and UTM tile sets encoded in the GeoPackage. Clients successfully displayed the USGS data following the USGS TNM template.

6.2.1. Support for Layer Ordering

Layer ordering was defined using the order of encounter in the `gpkg_contents` table.

6.2.2. Support for Universal Transverse Mercator (UTM) tiles

The USGS displays its TNM data in UTM projections. To support the requirement for UTM display of not only vector, but raster data stored in the GeoPackage, raster map images supplied by the WMTS and WMS servers available at the USGS were warped from their source projection to UTM and stored in the generated GeoPackage. Storing the tiles in a selected UTM projection makes lookup and display straightforward for the client. The image below [UTM Raster Support](#) shows imagery raster and shaded relief raster layers displayed in a UTM projection. The rasters are blended by changing the alpha channel of the imagery layer in the client.

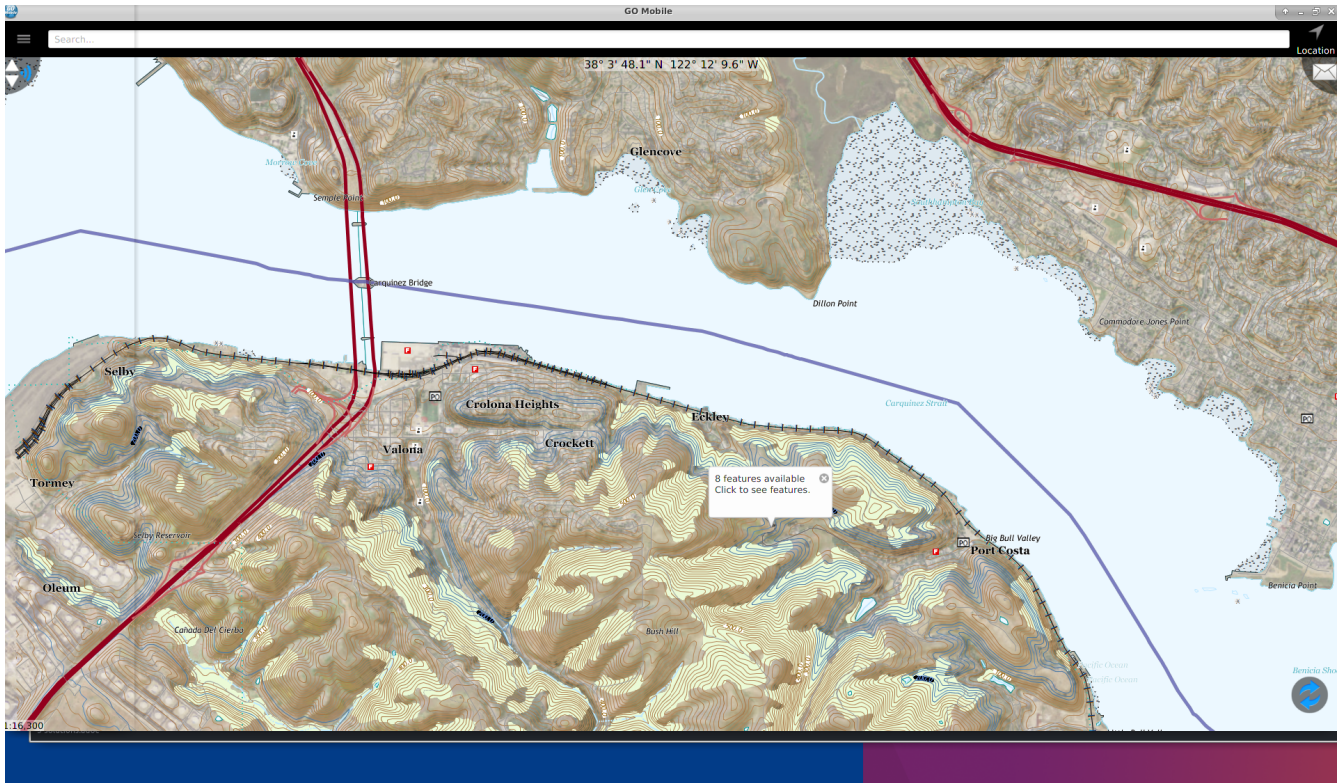


Figure 8. UTM Raster Support

6.2.3. Support for Dynamic Styling

As part of Testbed-13 a mobile client was used to interpret feature styling. This client can also be used to edit feature attributes. The side-effect of changing attributes is that the portrayal of features can change if the attributes are used to drive portrayal. GeoPackages created for this testbed include dynamic styling extensions that provide clients with enough information to re-interpret a feature's attributes and portray that feature differently if required. The example images below illustrate the editing of a fire station feature and re-portrayal as a school after the FCODE was changed. FCODE was the driving attribute affecting portrayal of this feature. The client recognized the change in the FCODE to trigger a re-interpretation of the styling rules and store a new style for that feature, that of a school.

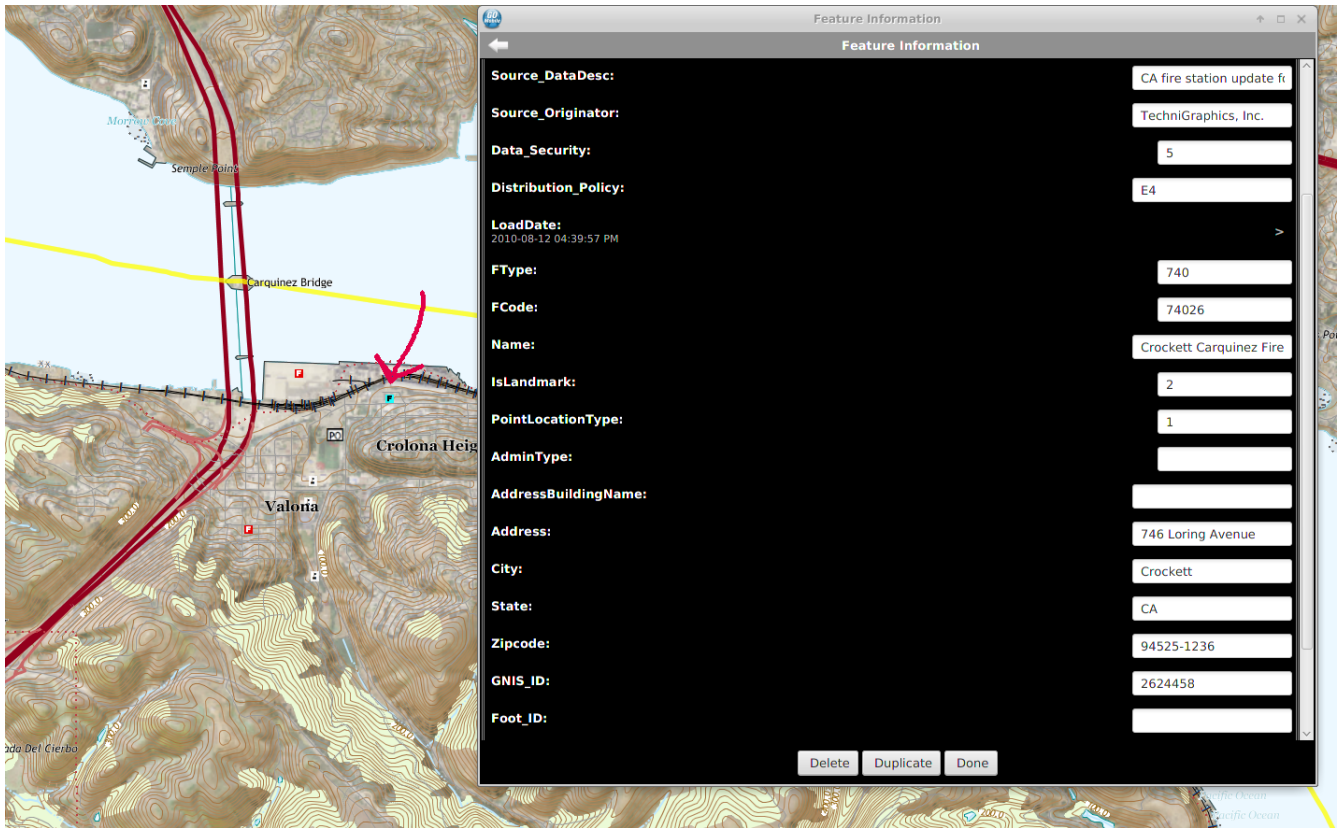


Figure 9. Editing A Fire Station Feature

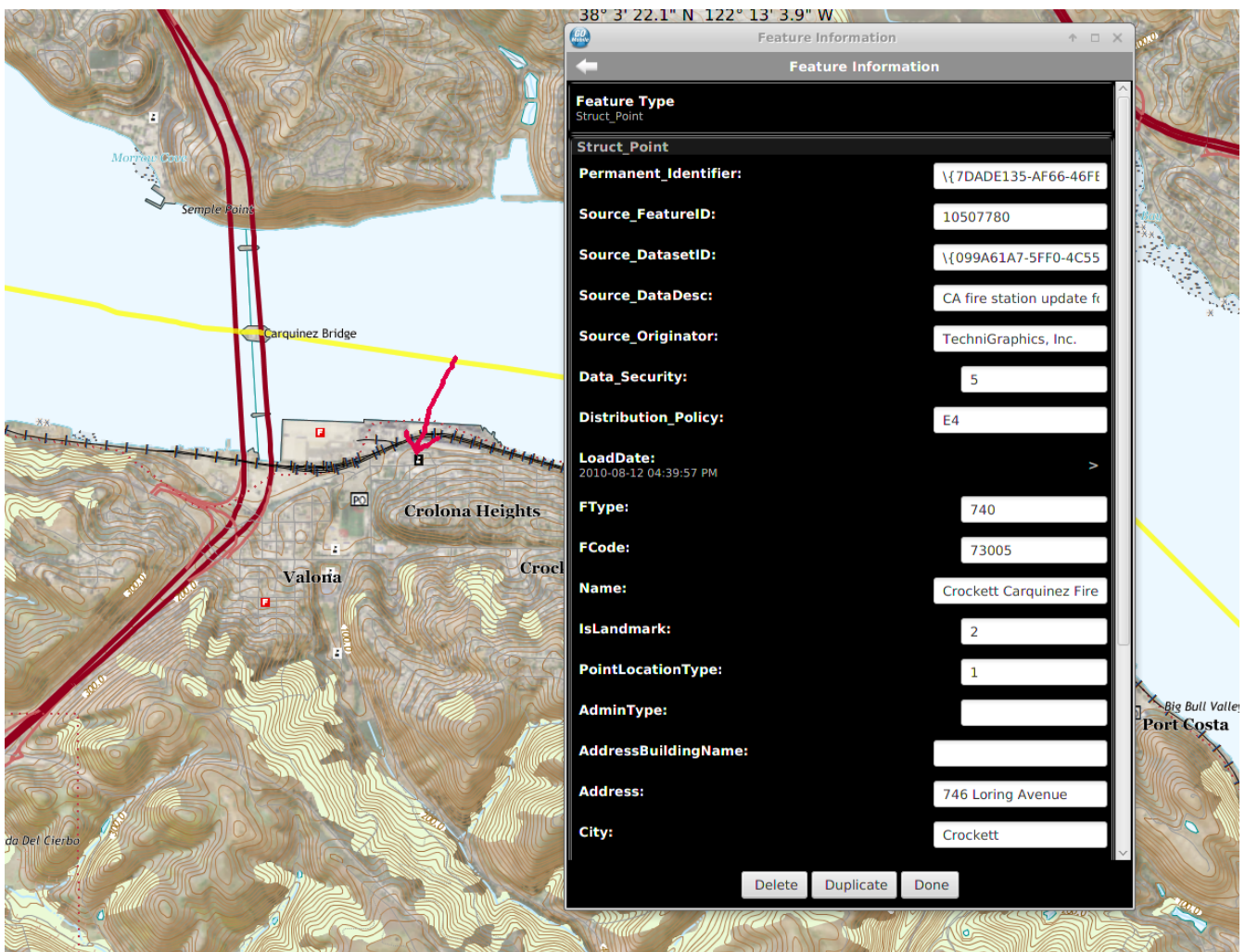


Figure 10. A Fire Station Becomes a School Feature

Appendix A: GeoPackage WPS Implementation

For Testbed-13, a WPS 2.0 GeoPackage generator was implemented. This WPS was a refinement of the one created for Testbed-12. There were two primary improvements :

1. Use a more immediate interface. The WPS in testbed 12 required registration of a number or items as a datasource as well as a stylesheet in order to begin generating the GeoPacakage. This required a learning curve and adoption of a mixture of JSON() and XML() which was not standards based. Clients interfacing with the WPS would be required to perform a number of calls to start the generation process. By simply using a context document, clients could send an "Execute" call directly and have the GeoPackage generated.
2. Align the input to OGC standards. Using a context document was beneficial because it is an adopted standard. There is an XML encoding for it which prevents the mixing and matching of JSON and XML which made the previous WPS more complicated. There are JSON encodings of context, but there is no JSON encoding standard for the WPS 2.0 interface.

The WPS is hosted at <https://ogc-tb13.compusult.net/wes/GeopackageWPS>. The endpoint is protected using HTTP Basic Authentication. A simple client has been provided to generate the requests and download the resulting GeoPackage. This client is a jar file that can be run from the command line. Command line arguments are required as follows :

```
JAVA_HOME/bin java -jar GPKGWPSClient.jar <config directory> <output directory>  
https://ogc-tb13.compusult.net/wes/GeopackageWPS username password
```

The executable jar file is available at : https://portal.opengeospatial.org/files/?artifact_id=76027. The accompanying configuration for the WPS client is available at https://portal.opengeospatial.org/files/?artifact_id=76028.

The form of the configuration directory is a directory of Java properties files that define input values used to construct the execute call. There is a global properties file, and a properties file for each context offering. Currently, the only offerings supported are WMS, WMTS and FileGDB. See below for examples. All properties are required.

A.1. Global Properties

The following list explains the global properties:

- offerings is a comma separated list of properties files that define the offerings contained in the context document.
- DESTSRS is the intended destination SRS for raster data stored in the GeoPackage
- SRCBBOX is a BBOX polygon defined in WKT. This BBOX defines the area from which raster data is harvested form the different offerings.
- SRCSRS is the SRS of the SRCBBOX polygon defined above

- Other parameters are used to define the content of the Context document.

The listing below illustrates an example of these properties.

```
template=shell.xml
offerings=wetlands.properties,beniciagdb.properties,imagery.properties,shadedrelief.pr
operties
DESTSRS=EPSG:32610
EMAIL=rcass@compusult.net
SITEURL=http://www.compusult.net
PUBLISHER=Compusult
GENERATOR=Compusult GeoPackager
SRCBBOX=POLYGON((-122.25 38, -122.25 38.125, -122.125 38.125, -122.125 38, -122.25 38
))
SRCSRS=EPSG:4326
TITLE=VECTOR 3330 Benicia 7 5 Min
AUTHOR=Compusult
```

A.2. WMS Offerings Properties

The following list explains the WMS properties:

- GETCAPABILITIES is the GetCapabilities request for the service in question
- GETMAP is an example reference GetMap call for the service in question
- ENTRY/OFFERING MIN/MAX SCALE are used to define the range of zoom levels harvested from the service in question
- STYLENAME and STYLETITLE are used to define the style referenced in the context document.
- TITLE is used to name the offering.
- CONTENT is simply the CONTENT of the offering
- FORMAT enforces the storage of the raster data

Even if the GetMap request is in a projection different from the DESTSRS defined in the global properties, the WPS will attempt to re-project the raster tiles acquired from the WMS to the DESTSRS.

The listing below illustrates an example of these properties.

```

template=wmsentry.xml
GETCAPABILITIES=https://fwspublicservices.wim.usgs.gov/server/services/Wetlands/MapServer/WmsServer?request=GetCapabilities&service=WMS
GETMAP=https://fwspublicservices.wim.usgs.gov/server/services/Wetlands/MapServer/WmsServer?SERVICE=WMS&VERSION=1.3.0&REQUEST=GetMap&WIDTH=800&HEIGHT=400&LAYERS=1&CRS=EPSG:4326&BBOX=-14.424688,-170.898427,71.43969,145.883474&FORMAT=image/png
ENTRYMINSCALE=1250
ENTRYMAXSCALE=400000
OFFERINGMINSCALE=1250
OFFERINGMAXSCALE=400000
STYLENAME=DEFAULT
STYLETITLE=DEFAULT
TITLE=Wetlands
CONTENT=This service cartographically renders the U.S. Fish and Wildlife Service wetlands and deepwater habitat data for use as a base layer wetland map information for general use products greater or equal to 1:100,000 scale. Three main categories were identified (Emergent Wetlands, Forest/Shrub Wetlands and Inland Waters) to render these features to resembles the USGS topographic maps. The emergent wetlands are symbolized with blue wetland map symbology and white polygon fill, without a polygon outline. The emergent wetlands depicted do not include lakes, rivers, open water ponds, deepwater marine and estuarine features or non-vegetated, farmed, intermittent and temporarily flooded wetlands. The forested and scrub/shrub wetlands are symbolized with blue wetland map symbology and green polygon fill, without a polygon outline. The forested and scrub/shrub wetlands depicted do not include lakes, rivers, open water ponds, deepwater marine and estuarine features or non-vegetated, farmed, intermittent and temporarily flooded wetlands. The inland waters include lakes, rivers, open water ponds, open water estuarine features and are symbolized with a light blue polygon fill, without a polygon outline. The water features depicted do not include vegetated or non-vegetated wetlands.
FORMAT=image/png

```

A.3. WMTS Offerings Properties

The following list explains the WMTS properties:

- GETCAPABILITIES is the GetCapabilities request for the service in question
- GETTILE is an example reference GetTile call for the service in question
- ENTRY/OFFERING MIN/MAX SCALE are used to define the range of zoom levels harvested from the service in question
- STYLENAME and STYLETITLE are used to define the style referenced in the context document.
- TITLE is used to name the offering.
- CONTENT is simply the CONTENT of the offering
- FORMAT enforces the storage of the raster data

The listing below illustrates an example of these properties.


```
template=wmtsentry.xml
GETCAPABILITIES=https://tnmbeta.cr.usgs.gov/arcgis/rest/services/USGSShadedRelief/MapServer/WMTS/1.0.0/WMTSCapabilities.xml?REQUEST=GetCapabilities&SERVICE=WMTS&VERSION=1.0.0
GETTILE=https://tnmbeta.cr.usgs.gov/arcgis/rest/services/USGSShadedRelief/MapServer/WMTS/tile/1.0.0/USGSShadedRelief/default/GoogleMapsCompatible/1/0/0.png
ENTRYMINSCALE=17061.83667079827
ENTRYMAXSCALE=500000
OFFERINGMINSCALE=17061.83667079827
OFFERINGMAXSCALE=500000
STYLENAME=DEFAULT
TITLE=USGS Shaded Relief
CONTENT=USGS Shaded Relief
FORMAT=image/png
```

A.4. FileGDB Offerings Properties

The following list explains the FileGDB properties:

- TITLE is used to name the offering.
- CONTENT is simply the CONTENT of the offering
- STYLENAME is the name of the style to be used to generate the symbology for the features in the FileGDB. The style must be available at the WPS.
- ENTRY/OFFERING MIN/MAX SCALE are used to define the scale ranges for display of the vector data contained in the FileGDB.
- GDBURL sets the URL used to download the FileGDB

The listing below illustrates an example of these properties.

```
template=gdbentry.xml
TITLE=USGS Combined Vector for Benicia, California 20160809 7.5 x 7.5 minute
CONTENT=Layers of geospatial data include contours, boundaries, land cover, hydrography, roads, transportation, geographic names, structures, and other selected map features.
STYLENAME=TNM
STYLETITLE=TNM
OFFERINGMINSCALE=0
OFFERINGMAXSCALE=400000
ENTRYMINSCALE=0
ENTRYMAXSCALE=400000
GDBURL=https://prd-tnm.s3.amazonaws.com/StagedProducts/Vector/GDB/VECTOR_3330_Benicia_7_5_Min.zip
```

A.5. WPS Operations

Most of the listed WPS operations require the use of a specialized URL. The name of the request is appended to the base URL of the WPS as in : <https://ogc-tb13.compusult.net/wes/GeopackageWPS/execute>.

A.5.1. Execute (Context Document)

The primary input for the WPS execute call is an OWS Context document encoded in XML. The context document is embedded in a ComplexData section of the execute call. An example is illustrated below. Because the generation process takes quite a while, it is recommended that the mode be set to 'async' which instructs the WPS to execute the process asynchronously.

```
<?xml version="1.0" encoding="UTF-8"?>
<wps:Execute xmlns:wps="http://www.opengis.net/wps/2.0" xmlns:ows=
"http://www.opengis.net/ows/2.0" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation=
"http://www.opengis.net/wps/2.0 ../wps.xsd" service="WPS" version="2.0.0" response=
"document" mode="async">
  <ows:Identifier>CreateGeoPackageViaContext</ows:Identifier>
  <wps:Input id="OWC_CONTEXT_DOCUMENT">
    <wps>Data>
      <wps:ComplexData>
        <wps:Format mimeType="application/xml"/>
        <feed xmlns="http://www.w3.org/2005/Atom" xml:lang="en">
          <link href="http://www.opengis.net/spec/owc-atom/1.0/req/core" rel="profile"
title="This file is compliant with version 1.0 of OWS Context"/>
          <id>3c70429f-b2a3-413c-82a8-9402684424dd</id>
          <title type="text">VECTOR 3330 Benicia 7 5 Min</title>
          <subtitle type="text">An ATOM record version of Compusult's OWS-
Context</subtitle>
          <updated>2017-10-01T05:14:06Z</updated>
          <author>
            <name>Compusult</name>
            <email>rcass@compusult.net</email>
            <uri>http://www.compusult.net</uri>
          </author>
          <publisher xmlns="http://purl.org/dc/elements/1.1/">Compusult</publisher>
          <generator uri="http://www.compusult.net" version="1.0">Compusult
GeoPackager</generator>
          <display xmlns="http://www.opengis.net/owc/1.0">
            <pixelWidth>1337</pixelWidth>
            <pixelHeight>876</pixelHeight>
          </display>
          <where xmlns="http://www.georss.org/georss">
            <Envelope xmlns="http://www.opengis.net/gml" srsDimension="2" srsName=
"EPSG:32610">
              <lowerCorner>
                <gml:Point srsDimension="2" srsName="EPSG:32610" xmlns:gml=
```

```

"http://www.opengis.net/gml">
    <gml:pos>565736.5196833616 4206080.364529469</gml:pos>
    </gml:Point>
</lowerCorner>
    <upperCorner>
    <gml:Point srsDimension="2" srsName="EPSG:32610" xmlns:gml=
"http://www.opengis.net/gml">
    <gml:pos>576823.3921379725 4220045.738070739</gml:pos>
    </gml:Point>
    </upperCorner>
</Envelope>
</where>
<entry>
    <id>9145ce5b-dbd8-4e5b-9c57-62ca9a100c28</id>
    <title type="text">Wetlands</title>
    <content type="html">This service cartographically renders the U.S. Fish
and Wildlife Service wetlands and deepwater habitat data for use as a base layer
wetland map information for general use products greater or equal to 1:100,000 scale.
Three main categories were identified (Emergent Wetlands, Forest/Shrub Wetlands and
Inland Waters) to render these features to resembles the USGS topographic maps. The
emergent wetlands are symbolized with blue wetland map symbology and white polygon
fill, without a polygon outline. The emergent wetlands depicted do not include lakes,
rivers, open water ponds, deepwater marine and estuarine features or non-vegetated,
farmed, intermittent and temporarily flooded wetlands. The forested and scrub/shrub
wetlands are symbolized with blue wetland map symbology and green polygon fill,
without a polygon outline. The forested and scrub/shrub wetlands depicted do not
include lakes, rivers, open water ponds, deepwater marine and estuarine features or
non-vegetated, farmed, intermittent and temporarily flooded wetlands. The inland
waters include lakes, rivers, open water ponds, open water estuarine features and are
symbolized with a light blue polygon fill, without a polygon outline. The water
features depicted do not include vegetated or non-vegetated wetlands. </content>
    <updated>2017-10-01T05:14:06Z</updated>
    <category scheme="http://www.opengis.net/spec/owc/active" term="true"/>
    <offering xmlns="http://www.opengis.net/owc/1.0" code=
"http://www.opengis.net/spec/owc-atom/1.0/req/wms">
    <operation code="GetCapabilities" href=
"https://fwspublicservices.wim.usgs.gov/server/services/Wetlands/MapServer/WmsServer?r
equest=GetCapabilities&service=WMS" method="GET" type="application/xml"/>
    <operation code="GetMap" href=
"https://fwspublicservices.wim.usgs.gov/server/services/Wetlands/MapServer/WmsServer?S
ERVICE=WMS&VERSION=1.3.0&REQUEST=GetMap&WIDTH=800&HEIGHT=400&LAYER
S=1&CRS=EPSG:4326&BBOX=-14.424688,-170.898427,71.43969,145.883474
&FORMAT=image/png" method="GET" type="image/png"/>
    <styleSet default="true">
    <name>DEFAULT</name>
    <title>Default</title>
    </styleSet>
</offering>
    <minScaleDenominator xmlns="http://www.opengis.net/owc/1.0">1250<
/minScaleDenominator>
    <maxScaleDenominator xmlns="http://www.opengis.net/owc/1.0">400000<

```

```

/maxScaleDenominator>
  <where xmlns="http://www.georss.org/georss">
    <Envelope xmlns="http://www.opengis.net/gml" srsDimension="2" srsName=
"EPSG:32610">
      <lowerCorner>
        <gml:Point srsDimension="2" srsName="EPSG:32610" xmlns:gml=
"http://www.opengis.net/gml">
          <gml:pos>565736.5196833616 4206080.364529469</gml:pos>
        </gml:Point>
      </lowerCorner>
      <upperCorner>
        <gml:Point srsDimension="2" srsName="EPSG:32610" xmlns:gml=
"http://www.opengis.net/gml">
          <gml:pos>576823.3921379725 4220045.738070739</gml:pos>
        </gml:Point>
      </upperCorner>
    </Envelope>
  </where>
</entry>
<entry>
  <id>8e892f05-9750-4436-a5f4-d28759915cb7</id>
  <title type="text">USGS Combined Vector for Benicia, California 20160809
7.5 x 7.5 minute</title>
  <content type="html">Layers of geospatial data include contours,
boundaries, land cover, hydrography, roads, transportation, geographic names,
structures, and other selected map features.</content>
  <updated>2017-10-01T05:14:06Z</updated>
  <category scheme="http://www.opengis.net/spec/owc/active" term="true"/>
  <offering xmlns="http://www.opengis.net/owc/1.0" code=
"http://www.opengis.net/spec/owc-atom/1.0/req/geodatabase">
    <content xmlns="http://www.opengis.net/owc/1.0" type="applicaton/x-zip-
compressed-ogc-gdb" href="https://prd-
tnm.s3.amazonaws.com/StagedProducts/Vector/GDB/VECTOR_3330_Benicia_7_5_Min.zip"/>
    <StyleSet default="true">
      <name>TNM</name>
      <title>TNM</title>
    </StyleSet>
  </offering>
  <minScaleDenominator xmlns="http://www.opengis.net/owc/1.0">0<
/minScaleDenominator>
  <maxScaleDenominator xmlns="http://www.opengis.net/owc/1.0">400000<
/maxScaleDenominator>
  <where xmlns="http://www.georss.org/georss">
    <Envelope xmlns="http://www.opengis.net/gml" srsDimension="2" srsName=
"EPSG:32610">
      <lowerCorner>
        <gml:Point srsDimension="2" srsName="EPSG:32610" xmlns:gml=
"http://www.opengis.net/gml">
          <gml:pos>565736.5196833616 4206080.364529469</gml:pos>
        </gml:Point>
      </lowerCorner>

```

```

        <upperCorner>
          <gml:Point srsDimension="2" srsName="EPSG:32610" xmlns:gml="
http://www.opengis.net/gml">
            <gml:pos>576823.3921379725 4220045.738070739</gml:pos>
          </gml:Point>
        </upperCorner>
      </Envelope>
    </where>
  </entry>
<entry>
  <id>bf2255e0-d0de-4d78-9b66-96b29119a0d6</id>
  <title type="text">USGSImageryOnly</title>
  <author>
    <name>Compusult</name>
  </author>
  <category scheme="http://www.opengis.net/spec/owc/active" term="true"/>
  <minScaleDenominator xmlns="http://www.opengis.net/owc/1.0">18000<
/minScaleDenominator>
  <maxScaleDenominator xmlns="http://www.opengis.net/owc/1.0">500000<
/maxScaleDenominator>
  <content type="html">USGSImageryOnly</content>
  <offering xmlns="http://www.opengis.net/owc/1.0" code="
http://www.opengis.net/spec/owc-atom/1.0/req/wmts">
    <operation code="GetCapabilities" href="
https://basemap.nationalmap.gov/arcgis/rest/services/USGSImageryOnly/MapServer/WMTS/1
.0.0/WMTSCapabilities.xml" method="GET" type="application/xml"/>
    <operation code="GetTile" href="
https://basemap.nationalmap.gov/arcgis/rest/services/USGSImageryOnly/MapServer/WMTS/t
ile/1.0.0/USGSImageryOnly/default/default028mm/0/0/0.png" method="GET" type="
image/png"/>
    <styleSet default="true">
      <name>DEFAULT</name>
      <title>Default</title>
    </styleSet>
  </offering>
  <where xmlns="http://www.georss.org/georss">
    <Envelope xmlns="http://www.opengis.net/gml" srsDimension="2" srsName=
"EPSG:32610">
      <lowerCorner>
        <gml:Point srsDimension="2" srsName="EPSG:32610" xmlns:gml="
http://www.opengis.net/gml">
          <gml:pos>565736.5196833616 4206080.364529469</gml:pos>
        </gml:Point>
      </lowerCorner>
      <upperCorner>
        <gml:Point srsDimension="2" srsName="EPSG:32610" xmlns:gml="
http://www.opengis.net/gml">
          <gml:pos>576823.3921379725 4220045.738070739</gml:pos>
        </gml:Point>
      </upperCorner>
    </Envelope>
  </where>

```

```

    </where>
  </entry>
  <entry>
    <id>f845de8b-e6d8-4630-a695-80aeb6d2c580</id>
    <title type="text">USGS Shaded Relief</title>
    <author>
      <name>Compusult</name>
    </author>
    <category scheme="http://www.opengis.net/spec/owc/active" term="true"/>
    <minScaleDenominator xmlns="http://www.opengis.net/owc/1.0"
>17061.83667079827</minScaleDenominator>
    <maxScaleDenominator xmlns="http://www.opengis.net/owc/1.0">500000<
/maxScaleDenominator>
    <content type="html">USGS Shaded Relief</content>
    <offering xmlns="http://www.opengis.net/owc/1.0" code=
"http://www.opengis.net/spec/owc-atom/1.0/req/wmts">
      <operation code="GetCapabilities" href=
"http://tnmbeta.cr.usgs.gov/arcgis/rest/services/USGSShadedRelief/MapServer/WMTS/1.0.
0/WMTSCapabilities.xml?REQUEST=GetCapabilities&SERVICE=WMTS&VERSION=1.0.0"
method="GET" type="application/xml"/>
      <operation code="GetTile" href=
"http://tnmbeta.cr.usgs.gov/arcgis/rest/services/USGSShadedRelief/MapServer/WMTS/tile
/1.0.0/USGSShadedRelief/default/GoogleMapsCompatible/1/0/0.png" method="GET" type=
"image/png"/>
      <styleSet default="true">
        <name>DEFAULT</name>
        <title>Default</title>
      </styleSet>
    </offering>
    <where xmlns="http://www.georss.org/georss">
      <Envelope xmlns="http://www.opengis.net/gml" srsDimension="2" srsName=
"EPSG:32610">
        <lowerCorner>
          <gml:Point srsDimension="2" srsName="EPSG:32610" xmlns:gml=
"http://www.opengis.net/gml">
            <gml:pos>565736.5196833616 4206080.364529469</gml:pos>
          </gml:Point>
        </lowerCorner>
        <upperCorner>
          <gml:Point srsDimension="2" srsName="EPSG:32610" xmlns:gml=
"http://www.opengis.net/gml">
            <gml:pos>576823.3921379725 4220045.738070739</gml:pos>
          </gml:Point>
        </upperCorner>
      </Envelope>
    </where>
  </entry>
</feed>
</wps:ComplexData>
</wps:Data>
</wps:Input>

```

```
<wps:Output id="GENERATION_STATUS">
</wps:Output>
</wps:Execute>
```

A.5.2. GetStatusRequest (JobId)

When running a WPS request in asynchronous mode, clients use the GetStatus request to poll the WPS to determine the status of the submitted Job. The JobId in the initial StatusInfo response from the initial execute request is used to poll the server.

```
<?xml version="1.0" encoding="UTF-8"?>
<wps:GetStatus service="WPS" version="2.0.0" xmlns:wps="
http://www.opengis.net/wps/2.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wps/2.0 ../wpsStatusInfo.xsd">
  <wps:JobID>6a7f68be-46c1-4f7d-8009-67ef0d5a8161</wps:JobID>
</wps:GetStatus>
```

A.5.3. StatusInfo Response (JobId)

StatusInfo responses contain Failure, Processing or Success information.

```
<?xml version="1.0" encoding="UTF-8"?>
<StatusInfo xmlns="http://www.opengis.net/wps/2.0" xmlns:xsi=
"http://www.w3.org/2001/XMLSchema-instance">
  <JobID>6a7f68be-46c1-4f7d-8009-67ef0d5a8161</JobID>
  <Status>Accepted</Status>
  <ExpirationDate xsi:nil="true"/>
  <EstimatedCompletion xsi:nil="true"/>
  <NextPoll>2017-10-01T00:00:00.000-02:30</NextPoll>
  <PercentCompleted>0</PercentCompleted>
</StatusInfo>
```

An example of a Failed Job.

```
<?xml version="1.0" encoding="UTF-8"?>
<StatusInfo xmlns="http://www.opengis.net/wps/2.0" xmlns:xsi=
"http://www.w3.org/2001/XMLSchema-instance">
  <JobID>3d165ae6-67d3-4262-8bc7-44313940d0a4</JobID>
  <Status>Failed</Status>
  <ExpirationDate xsi:nil="true"/>
  <EstimatedCompletion xsi:nil="true"/>
  <NextPoll>2017-10-02T00:00:00.000-02:30</NextPoll>
  <PercentCompleted>0</PercentCompleted>
</StatusInfo>
```

A.5.4. GetResult Request (JobId)

A client gets a Successful status after polling the job repeatedly. At this point, the generated GeoPackage can be downloaded.

```
<?xml version="1.0" encoding="UTF-8"?>
<StatusInfo xmlns="http://www.opengis.net/wps/2.0" xmlns:xsi=
"http://www.w3.org/2001/XMLSchema-instance">
  <JobID>984ac8e5-21d7-461e-8b01-c7fc5a7faba8</JobID>
  <Status>Succeeded</Status>
  <ExpirationDate xsi:nil="true"/>
  <EstimatedCompletion>2017-10-03T00:00:00.000-02:30</EstimatedCompletion>
  <NextPoll>2017-10-03T00:00:00.000-02:30</NextPoll>
  <PercentCompleted>100</PercentCompleted>
</StatusInfo>
```

A client retrieves the download URL for a successfully completed GeoPackage using a GetResult call.

```
<?xml version="1.0" encoding="UTF-8"?>
<wps:GetResult service="WPS" version="2.0.0" xmlns:wps="
http://www.opengis.net/wps/2.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wps/2.0 ../wpsGetResult.xsd ">
  <wps:JobID>984ac8e5-21d7-461e-8b01-c7fc5a7faba8</wps:JobID>
</wps:GetResult>
```

The client can use the URL in the data component to download the completed GeoPackage.

```
<?xml version="1.0" encoding="UTF-8"?>
<Result xmlns="http://www.opengis.net/wps/2.0" xmlns:xsi=
"http://www.w3.org/2001/XMLSchema-instance">
  <JobID>984ac8e5-21d7-461e-8b01-c7fc5a7faba8</JobID>
  <ExpirationDate xsi:nil="true"/>
  <Output>
    <Data mimeType="text/plain">https://ogc-tb13.compusult.net/wes/gpkg/gp/22273860-
e18e-4030-8cfc-de70each3ee8</Data>
  </Output>
</Result>  auth string: wes:wes
```


Appendix B: Static Feature Symbology Extension (Normative)

B.1. Extension Title

The title of this extension is Static Feature Symbology.

B.2. Introduction

While feature data stored in GeoPackages can be visually represented by client applications in an infinite number of ways, producers of GeoPackage products containing feature data often wish to have these products depicted in a consistent manner based on rules that govern rendering, such as color, size, orientation, icons, etc.

In order to support this requirement, this extension is provided to store these symbologies and their application to features stored in GeoPackage feature tables.

The extension is designed to facilitate association of individual features with an appropriate symbolization, based on viewing scale, to determine an appropriate symbology.

The symbolization is "hard-coded" for the feature. Each feature to be rendered should be linked to a symbology created by the producer and stored in the GeoPackage at creation time.

The styling information for any feature table is linked to a "styling package." This package is registered in a table called `gpkg_extended_contents`.

B.3. Extension Author

The author of this extension is Compusult Limited.

B.4. Extension Template

For each feature table requiring extension for symbology, the following extended tables shall be created and registered in the `gpkg_extensions` table:

- `<style_package>_style_rules`
- `<style_package>_styles`
- `<style_package>_style_images`
- `<style_package>_style_links`

The `style_package` prefix is determined by an entry in the `gpkgext_extended_contents` table, which links a feature table name to a `style_pkg` defined in the `style_pkg` column, e.g.:

Table 2. *gpkgext_extended_contents* Table Example Row

table_name	max_scale_denom	style_pkg	identifiable
structures	10000	structures	1

The `style_pkg` column shows that a style package called "structures" is used for a table called "structures".

B.5. Extension Type

New requirement dependent on Clause 2.1.

B.6. Applicability

This extension applies to tables defined under Clause 2.1, i.e., vector features.

B.7. Scope

This is a read-write extension such that clients may only read from associated symbology to render, or clients may trigger the update of tables that contain the associated style/feature mappings with new symbology.

B.8. Requirements

B.8.1. Table Definitions

Extended Contents

Requirement 1

A GeoPackage that conforms to this extension shall, for each feature table covered by this extension, be registered in a table called `gpkg_extended_contents`. The `style_package` is linked on a 1:1 basis to the layer in this table. It is typically the name of the feature table to which it applies, e.g., a style package for a "roads" table would be called roads. This table shall be registered in `gpkg_extensions` with the `extension_name` as 'compusult_extcontents1' a definition value of 'TBD' and `column_name` value of NULL.

Table 3. *gpkg_extended_contents* Table

Column Name	Data Type	Description	Key
table_name	TEXT	The core table name to which these contents apply.	PK
max_scale_denom	REAL	The maximum scale denominator for which this table's features can be displayed.	
style_pkg	TEXT	A package name that links the feature table to its corresponding style package tables that are prefixed with this package name.	
identifiable	INTEGER	Boolean (0 or 1) indicating that the feature should be "identified" by clients.	

Extended Contents Table Definition SQL (Normative)

```
CREATE TABLE gpkgext_extended_contents
(
  table_name TEXT NOT NULL PRIMARY KEY,
  max_scale_denom REAL,
  style_pkg TEXT,
  identifiable INTEGER DEFAULT 1,
  CONSTRAINT gpkg_extended_contents_fk01 FOREIGN KEY (table_name) REFERENCES
  gpkg_contents(table_name)
);
```

Styles

Requirement 2

A GeoPackage that conforms to this extension shall, for each feature table covered by this extension, contain a table whose name is derived from the feature using the following template: `<style_package>_styles`. The style_package prefix is typically the name of the feature table to which it applies, e.g., a style package for a "roads" table would be called roads. This table shall be registered in `gpkg_extensions` with the extension_name as 'compusult_<style_package>_styles1b', a definition value of 'TBD' and column_name value of NULL.

Table 4. <style_package>_styles Table

Column Name	Data Type	Description	Key
id	INTEGER	The primary key of the style limited to values in the range [0, 9223372036854775807].	PK
ordinal_pos	INTEGER	Indicates the order of application for the style if there is more than one style rule applicable for a feature. Falls into the range [0, 9223372036854775807].	
rule_id	INTEGER	A positive number that is a foreign key to the corresponding rules table.	FK
style	TEXT	The encoded style or symbology instructions detailing how the feature should appear.	
style_encoding	TEXT	A mime type that indicates how the style is encoded, e.g., "application/vnd.ogc.se_xml".	
priority	REAL	A priority used to weight the symbolization of the feature used to remove cluttering.	

Styles Table Definition SQL (Normative)

```
CREATE TABLE <style_package>_styles
(
  id          INTEGER PRIMARY KEY AUTOINCREMENT,
  ordinal_pos INTEGER NOT NULL,
  rule_id     INTEGER NOT NULL,
  style       TEXT,
  style_encoding TEXT,
  priority    REAL,
  CONSTRAINT <style_package>_styles_fk01
  FOREIGN KEY (rule_id) REFERENCES <style_package>_style_rules (id) ON
  DELETE CASCADE
);
```

Style Rules

Requirement 3

A GeoPackage that conforms to this extension shall, for each feature table covered by this extension, contain a table whose name is derived from the feature using the following template: `<style_package>_style_rules`. This table shall be registered in `gpkg_extensions` with the `extension_name` as `'compusult_<style_package>_styles1a'` and definition of 'TBD' and `column_name` of NULL.

Table 5. `<style_package>_style_rules` Table

Column Name	Data Type	Description	Key
id	INTEGER	The primary key of the rule limited to values in the range [0, 9223372036854775807].	PK
ordinal_pos	INTEGER	Indicates the order of application for the rule if there is more than one style rule applicable for a feature. Falls into the range [0, 9223372036854775807].	
min_scale_denom	REAL	A real value used for evaluating when the rule should be applied to a feature. When the scale denominator of the visible map is $>$ min_scale_denom, the rule is active.	
max_scale_denom	REAL	A real value used for evaluating when the rule should be applied to a feature. When the scale denominator of the visible map is $<$ max_scale_denom, the rule is active.	
is_else	INTEGER	A real value used for evaluating when the rule is applied if no other rules have been applied to the feature. 0 means that the rule should be applied, 1 means that it should not.	

Column Name	Data Type	Description	Key
rule_type	INTEGER	0 means that the style is for a geometry. 1 means that the style is a label style.	

Style_Rules Table Definition SQL (Normative)

```
CREATE TABLE <style_package>_style_rules
(
  id          INTEGER PRIMARY KEY AUTOINCREMENT,
  ordinal_pos INTEGER NOT NULL,
  min_scale_denom REAL NOT NULL,
  max_scale_denom REAL NOT NULL,
  is_else     INTEGER NOT NULL,
  rule_type   INTEGER NOT NULL
);
```

Style Images

Requirement 4

A GeoPackage that conforms to this extension shall, for each feature table covered by this extension, contain a table whose name is derived from the feature using the following template: `<style_package>_style_images`. This table shall be registered in `gpkg_extensions` with the `extension_name` as `'compusult_<style_package>_styles1c'` a definition value of `'TBD'` and `column_name` value of `NULL`.

Table 6. `<style_package>_style_images` Table

Column Name	Data Type	Description	Key
id	INTEGER	The primary key of the image limited to values in the range [0, 9223372036854775807].	PK
path	TEXT	The original path of the symbol, such as <code>"/usr/share/images/tower.png"</code> .	
data	BLOB	Binary data.	

Style_Images Table Definition SQL (Normative)

```
CREATE TABLE <style_package>_style_images
(
  id INTEGER PRIMARY KEY AUTOINCREMENT,
  path TEXT UNIQUE,
  data BLOB
);
```

Style Fonts

Requirement 4

A GeoPackage that conforms to this extension shall, contain a table whose name is : `gpkgext_custom_fonts`. This table shall be registered in `gpkg_extensions` with the `extension_name` as 'compusult_custom_fonts' a definition value of 'TBD' and `column_name` value of NULL.

Table 7. `gpkgext_custom_fonts` Table

Column Name	Data Type	Description	Key
identifier	text	The primary key of the font. This is also the font typeface name such as 'Helvetica'	PK
min_size	REAL	The minimum point size possible for this font	
max_size	REAL	The maximum point size possible for this font	
supports_bold	INTEGER	1 if the font file supports bold text	
supports_italic	INTEGER	1 if the font file supports italic text	
font	BLOB	Binary data (truetype font file).	

gpkgext_custom_fonts Table Definition SQL (Normative)

```
CREATE TABLE gpkgext_custom_fonts (  
  identifier text primary key,  
  min_size real,  
  max_size real,  
  supports_bold integer,  
  supports_italic integer,  
  font blob  
)
```

Style Links

Requirement 5

A GeoPackage that conforms to this extension shall, for each feature table covered by this extension, contain a table whose name is derived from the feature using the following template: `<style_package>_style_links`. This table shall be registered in `gpkg_extensions` with the `extension_name` as `'compusult_<style_package>_styles1d'` and definition of 'TBD' and `column_name` of NULL.

Table 8. `<style_package>_style_links`

Column Name	Data Type	Description	Key
feature_id	INTEGER	A foreign key to a feature in the related feature table.	FK
rule_id	INTEGER	A foreign key to a rule related rule table.	FK

Style_Links Table Definition SQL (Normative)

```
CREATE TABLE <style_package>_style_links  
(  
  feature_id INTEGER NOT NULL,  
  rule_id   INTEGER NOT NULL,  
  CONSTRAINT <style_package>_style_link_pk UNIQUE (feature_id, rule_id),  
  CONSTRAINT <style_package>_style_link_fk01 FOREIGN KEY (feature_id) REFERENCES  
<style_package> (id) ON  
  DELETE CASCADE,  
  CONSTRAINT <style_package>_style_link_fk02 FOREIGN KEY (rule_id) REFERENCES  
<style_package> (id) ON  
  DELETE CASCADE  
);
```


Appendix C: Dynamic Feature Symbology Extension (Normative)

In addition to supporting direct representational styling using [Static Feature Symbology Extension \(Normative\)](#), an additional extension is provided to support dynamic styling of features. Dynamic styling is applicable when the attributes of a feature that determine its appearance have changed. For example, a user changes a road attribute such as lanes from "2" to "4". Styling rules exist that require road features with a lane value of "4" to be rendered with a median. Unless this rule is dynamically interpreted, the road will not be updated visually unless there is a mechanism to find an applicable set of style rules for the changed feature and determine which one applies to the new condition. If such a mechanism exists, the applicable rule will be used to determine the new symbology. In the road case, the expression "lanes=4" would be encountered, the corresponding symbology would be set for the feature and clients would now render the feature with a median.

To support this dynamic mechanism, a further extension to the [Static Feature Symbology Extension \(Normative\)](#) is encoded to support the necessary links between symbology and rule expressions.

The core idea of the dynamic symbolization is that a set of expressions are evaluated for any feature in the feature table. If the match expression is satisfied, any style from the dynamic styles table that is linked to that expression is applied to the feature. Application of these styles could occur at run time, or they can be computed and stored in the data model for static styles. The following are detailed descriptions of the extended tables.

These extended tables are linked via a mapping in `gpkg_extended_contents` that maps the feature table to the corresponding style tables below. This mapping is a requirement of the static styling extension which this extension is dependent upon.

C.1. Extension Author

This extension was authored by Compusult Limited.

C.2. Extension Name or Template

For each feature table requiring symbology, the following extended tables shall be created and registered in the `gpkg_extensions` table:

- `<style_package>_dynamic_styles`
- `<style_package>_dynamic_styles_images`
- `<style_package>_dynamic_styles_expressions`

The style package name is directly related to the style package registered for the complementary static style tables registered in the `gpkg_extended_contents` table.

C.3. Extension Type

New requirement dependent on Clause 2.1.

C.4. Applicability

This extension applies to tables defined under Clause 2.1, i.e., vector features. Additionally, it is dependent on the Static Styling Extension detailed in this document.

C.5. Scope

This is a read-only extension to support clients editing the attributes of features associated with the symbology which triggers the update of static style tables that contain the associated style/feature mappings.

C.6. Requirements

C.6.1. Table Definitions

Requirement 1

A GeoPackage that conforms to this extension shall, for each feature table covered by this extension, contain a table whose name is derived from the feature table name using the following template: `<style_package>_dynamic_styles`. This table shall be registered in `gpkg_extensions` with the extension_name as `compusult_<style_package>_dynamic_styles1b` and definition of `TBD` and column_name of `NULL`. The `_dynamic_styles` table is used to store feature symbology information and that information's relation to a boolean expression that determines when that rule should be applied to a feature. An example would be a style for a red road line. A line feature from `<tablename>` would get its symbology (red line) from this table. Additionally, this table would dictate which style would be applied based on a combination of the scale denominator range for that style and a successful evaluation of that style's related boolean expression based on a feature's attributes.

Table 9. `<style_package>_dynamic_styles`

Column Name	Data Type	Description	Key
id	INTEGER	The primary key of the rule limited to values in the range [0, 9223372036854775807].	PK
expression_id	INTEGER	A foreign key to the <code>dynamic_style_expressions</code> table for the feature table.	FK

Column Name	Data Type	Description	Key
min_scale_denom	REAL	A real value used for evaluating when the style should be applied to a feature. When the scale denominator of the visible map is > min_scale_denom, the style is active.	
max_scale_denom	REAL	A real value used for evaluating when the style should be applied to a feature. When the scale denominator of the visible map is < max_scale_denom, the style is active.	
style	TEXT	The text encoding of the symbology information, for example, an OGC symbology encoding document.	
style_encoding	TEXT	The mime type of the style encoding, such as "application/vnd.ogc.se_xml".	
priority	REAL	Indicates the order of application for the styles if there are multiple styles applicable for a feature.	

Dynamic_Styles Table Definition SQL (Normative)

```
CREATE TABLE <style_package>_dynamic_styles
(
  id                INTEGER PRIMARY KEY,
  expression_id    INTEGER,
  ordinal_pos      INTEGER,
  min_scale_denominator REAL,
  max_scale_denominator REAL,
  style            text,
  style_encoding   text,
  priority         REAL
);
```

Requirement 2

A GeoPackage that conforms to this extension shall, for each feature table covered by this extension, contain a table whose name is derived from the feature table name using the following template: `<style_package>_dynamic_style_expressions`. This table shall be registered in `gpkg_extensions` with the `extension_name` as `'compusult_<style_package>_dynamic_styles1a'` and definition of 'TBD' and `column_name` of NULL. The `_dynamic_style_expressions` table holds expressions used to evaluate a feature based on its attributes. If a feature successfully matches an expression, that feature is styled based on the styles that are related to that expression.

Table 10. `<style_package>_dynamic_style_expressions`

Column Name	Data Type	Description	Key
id	INTEGER	A foreign key to a feature in the related feature table.	PK
match_expr	TEXT	A SQL expression applied to attributes of the feature; it can be used directly in queries against the feature table.	
precedence	INTEGER	This value determines the order of evaluation for all match expressions associated with a feature.	

Dynamic_Style_expressions Table Definition SQL (Normative)

```
CREATE TABLE <style_package>_dynamic_style_expressions
(
  id INTEGER PRIMARY KEY,
  match_expr text,
  precedence INTEGER
);
```

Requirement 3

A GeoPackage that conforms to this extension shall, for each feature table covered by this extension, contain a table whose name is derived from the feature table name using the following template: `<style_package>_dynamic_style_images`. This table shall be registered in `gpkg_extensions` with the `extension_name` as `'compusult_<style_package>_dynamic_styles1c'` and definition of 'TBD' and `column_name` of NULL. Many symbology encodings reference external image files. In order for the image files to be accessible in a portable and unconnected manner, these images must be stored in the database and have a mechanism for referencing the images. The `_dynamic_style_images` table stores the images in data column. The symbology encoding in the `_dynamic_styles` table shall use the id in the `_dynamic_style_images` table to refer to the image.

Table 11. `<style_package>_dynamic_style_images`

Column Name	Data Type	Description	Key
id	INTEGER	The primary key of the image limited to values in the range [0, 9223372036854775807].	PK
path	TEXT	The original path of the symbol, such as <code>"/usr/share/images/tower.png"</code> .	
data	BLOB	Binary data.	
mime_type	TEXT	Mime type of the stored image such as <code>"image/png"</code> .	

Dynamic_Style_images Table Definition SQL (Normative)

```
CREATE TABLE <style_package>_dynamic_style_images
(
  id INTEGER PRIMARY KEY,
  path text,
  data BLOB,
  mime_type text
);
```

Appendix D: UTM Tile Matrices

This section describes a standardized approach to organizing tile matrices in a UTM projection to complement well-known global tile matrices such as World Mercator, Google Maps Compatible etc.

NOTE

For Testbed-13, the participants adopted the approach explained here. During the testbed, an alternative tile matrix approach based on a projected space of the whole planet was proposed. This new proposal is being reviewed by various working groups and may be adopted in the future. Due to time constraints, this approach was not coded into the solution used for the testbed.

UTM presents a challenge as too much measurement distortion occurs in areas to the east and west of a zone. The usable area covered by a rectangular tile matrix set that encompasses the UTM zone is an order taller than its width. Even though this matrix set is rectangular, the areas closer to the poles outside the zone contain areas of greater than acceptable distortion as the standard bounds of the designated zone do not form a rectangle as seen in the diagram below.

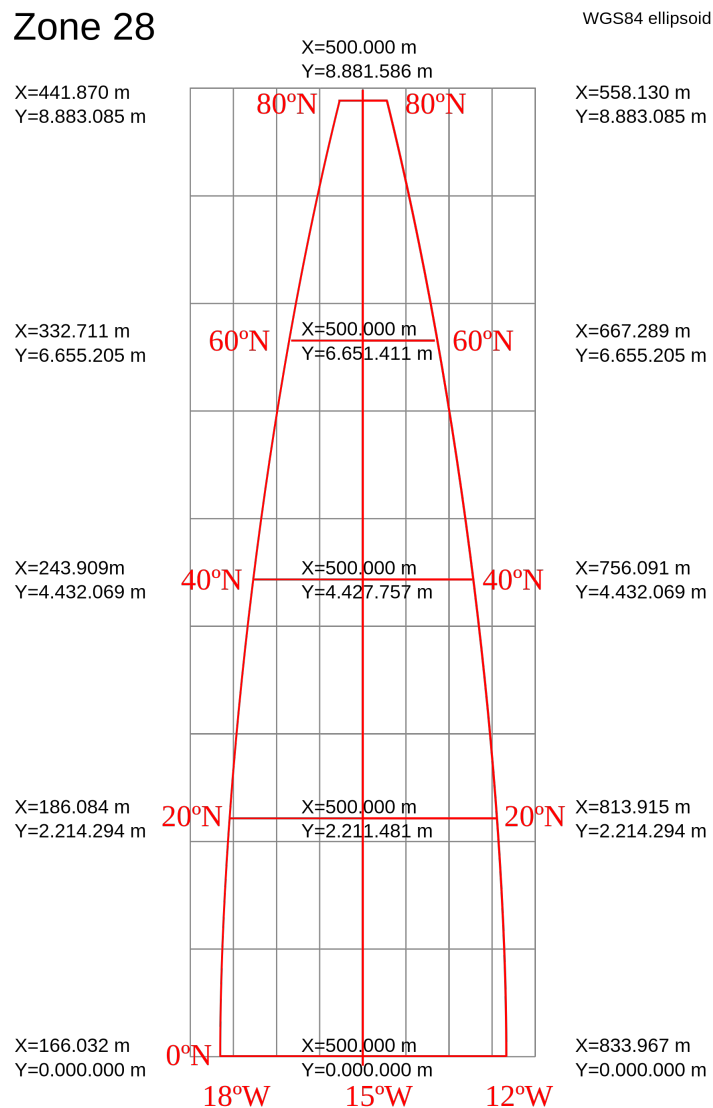


Figure 11. UTM Zone - northern hemisphere © Wikimedia Commons

UTM zones have defined bounds for which the data is reliable, but information beyond these

bounds has reasonable reliability to a point. The goal for a tile matrix set is to include a reasonable buffer so that adjacent data can be visualized in that zone, the buffer should be restrictive enough to prevent highly distorted data from being visualized.

Real world uses of UTM in digital mapping require data rendered in a single zone, even though the data may be outside the standard area of the zone.

To accommodate this overlap, some compromise has to be made between distortion and practicality.

Sources indicate that the acceptable "overlap" extent is 1/2 degree on either side of the 6 degree zone. In equatorial degrees, 1/2 degree is $111,395/2$ or 55607.5 meters.

Additionally, there is a latitude overlap that should be considered as well. While the standard UTM zone is limited at a latitude, 84 in the north, the projection algorithm will calculate beyond these latitudes allowing overlap here as well.

D.1. Calculating a usable extent

The easiest way to calculate a usable extent is to calculate the maximum bounding box using the zone extents plus overlap. For example for EPSG:32631 WGS 84 / UTM zone 31N , the calculated zone (with offset) corner points are :

-0.5, -0.5 \Rightarrow 110308.33, -55369.00

6.5, -0.5 \Rightarrow 889691.67, -55369.00

-0.5, 80.5 \Rightarrow 435547.76, 8939336.50

6.5, 80.5 \Rightarrow 564452.24, 8939336.50

The results above indicate that the northern-most longitudinal extent (128904.48) of the zone with overlap is almost 1/6th the southernmost longitudinal extent which is 779383.34.

Tile matrix bounds for WMTS etc. are rectangles, and the calculated UTM values do not describe a rectangle. Interpolated, they would reflect the extents in the diagram.

To meet this rectangular requirement, the matrix bounding rectangle will include data outside the zone. Thus the bounding rectangle gets extended to the maximum bounding rectangle for all four points :

110308.33, -55369.00

110308.33, 8939336.50

889691.67, 8939336.50

889691.67, -55369.00

This creates a rectangle whose aspect ratio is $(8939336.50 / (889691.67 - 110308.33)) = 11.469755692$.

This value is not a whole number. A whole number aspect ratio is required for an exact number of square tiles to fit. The current width of the rectangle is 779383.34. Flooring the ratio and keeping the latitude extents yields a new width of $8939336.50/11 = 812666.954545455$. The difference being old width and new width being $(812666.954545455 - 779383.34)$ or 33283.614545455. This would support a level 0 tile matrix that was 11 tiles high, implying we extend the width by $33283.614545455/2$ in both directions.

Applying half this difference on both sides of the zone to a northern UTM zone with overlaps creates a new bounding box of :

93666.522727273, -55369.00 \Rightarrow 0.6492683°, -0.499918°

93666.522727273, 8939336.50 \Rightarrow -18.0979403°, 79.8495941°

906333.477272728, 8939336.50 \Rightarrow 24.0979403°, 79.8495941°

906333.477272728, -55369.00 \Rightarrow 6.6492683°, -0.4999181°

D.2. A UTM Example

The following tables show an example of a GeoPackage TileMatrixSet using this tile matrix approach. The overlap used in this example was set at 100000m and not 0.5 degrees:

Table 12. TileMatrix Set

table_name	srs_id	min_x	min_y	max_x	max_y
VECTOR3330Benicia75Min_15a7ce10_1abd_4883_8168_5c50210bde43	32610	66021.4430960779	-109261.534717933	933978.556903925	9438266.71716838

Table 13. TileMatrix Definition

table_name	zoom_level	matrix_width	matrix_height	tile_width	tile_height	pixel_x_size	pixel_y_size
VECTOR3330Benicia75Min_15a7ce10_1abd_4883_8168_5c50210bde43	5	32	352	256	256	105.951796119122	105.951796119122
VECTOR3330Benicia75Min_15a7ce10_1abd_4883_8168_5c50210bde43	6	64	704	256	256	52.975898059561	52.975898059561

table_name	zoom_level	matrix_width	matrix_height	tile_width	tile_height	pixel_x_size	pixel_y_size
VECTOR330Benicia75Min_15a7ce10_1abd_4883_8168_5c50210bde43	7	128	1408	256	256	26.4879490297805	26.4879490297805
VECTOR330Benicia75Min_15a7ce10_1abd_4883_8168_5c50210bde43	8	256	2816	256	256	13.2439745148902	13.2439745148902
VECTOR330Benicia75Min_15a7ce10_1abd_4883_8168_5c50210bde43	9	512	5632	256	256	6.62198725744512	6.62198725744512
VECTOR330Benicia75Min_15a7ce10_1abd_4883_8168_5c50210bde43	10	1024	11264	256	256	3.31099362872256	3.31099362872256
VECTOR330Benicia75Min_15a7ce10_1abd_4883_8168_5c50210bde43	11	2048	22528	256	256	1.65549681436128	1.65549681436128
VECTOR330Benicia75Min_15a7ce10_1abd_4883_8168_5c50210bde43	12	4096	45056	256	256	0.82774840718064	0.82774840718064

table_name	zoom_level	matrix_width	matrix_height	tile_width	tile_height	pixel_x_size	pixel_y_size
VECTOR330Benicia75Min_15a7ce10_1abd_4883_8168_5c50210bde43	13	8192	90112	256	256	0.41387420359032	0.41387420359032
VECTOR330Benicia75Min_15a7ce10_1abd_4883_8168_5c50210bde43	14	16384	180224	256	256	0.20693710179516	0.20693710179516

Appendix E: Revision History

Table 14. Revision History

Date	Editor	Release	Primary clauses modified	Descriptions
Oct. 4th, 2017	R. Cass	.1	all	initial version
Oct. 31st, 2017	R. Cass	.2	all	DER version
Nov. 6th, 2017	R. Cass	.3	all	incorporated feedback from K. Fishburn, G. Hobona
Nov. 15th, 2017	R. Cass	.4	all	incorporated feedback from G. Buehler, prepared for pending

Appendix F: Bibliography

- [1] Strobel, S. et.al: DGIWG – Web Map Service 1.3 Profile, https://portal.opengeospatial.org/files/?artifact_id=66915 (2016)
- [2] Unknown, Department of The Army: FM 21-26 Map Reading, 1956
- [3] Bocher, E. and Ertz, O.: A redesign of OGC Symbology Encoding standard for sharing cartography, <https://peerj.com/preprints/2415.pdf>
- [4] Masó, Joan: OGC Tile Matrix Set Standard Candidate, https://portal.opengeospatial.org/files/?artifact_id=75045 (2017)