Open Geospatial Consortium

Training Session on Authoring OGC Standards with Metanorma

Part 2

Getting to know Metanorma for OGC

Meeting sponsor

MPA SINGAPORE

SLA SINGAPORE LAND AUTHORITY

GeoWorks SINGAPORE

2022-09-28

Overview

- How to add images
- How to add tables
- How to add code listings
- How to add mathematical formulae
- Specifying ModSpec elements (requirements classes, requirements, conformance classes, and conformance tests)
- Populating the bibliography and references sections

Introduction

- Part 1 of the training session introduced you to *Metanorma for OGC* by:
  - explaining the basics of metanorma
  - providing a hands-on exercise that installs metadata and compiles a document

- Part 2 will provide a hands-on exercise on how to add images, tables, code listings, mathematical formulae, as well as elements of OGC Standards and other specifications

Recap: Difference between asciidoctor and metanorma for OGC

Feature	Ascidoctor	Metanorma for OGC
Document metadata (e.g. docnumber, keywords, etc)	Added as values in table cells	Added as document attributes (More info)
Requirements Classes	Presentation and content specified as tables	Created using a definition list, and then automatically rendered as tables. (More info)
Requirements	Presentation and content specified as tables	Created using a definition list, and then automatically rendered as tables. (More info)
Conformance Classes	Presentation and content specified as tables	Created using a definition list, and then automatically rendered as tables. (More info)
Conformance Tests	Presentation and content specified as tables	Content added as a definition list, and then automatically rendered as tables for Presentation. (More info)

https://github.com/opengeospatial/templates/tree/master/cheatsheet

Hands-on Exercise

Recap: Compiling a draft OGC Standard with a docker-containerized Metanorma instance

- To convert the draft standard from asciidoc format to HTML and PDF formats, we use the metanorma software to compile the document.

1. From the folder containing the `document.adoc` file, run the following command.

```
docker run -v "$(pwd)":/metanorma -v
${HOME}/.fontist/fonts/:/config/fonts metanorma/metanorma
metanorma compile --agree-to-terms -t ogc -x
xml,html,doc,pdf document.adoc
```

How to add images to an AsciiDoc file

- Preferably use PNG encoded images
- Ideally place the images in the images folder (be consistent)
- Recommended width of 800 pixels or more
- Place an anchor (e.g. [[my_anchor]]) above the image so that it can be referenced
- Always add a caption (preceded by dot on the example below)

```
[[img_architecture]]
.High Level Overview of the Sprint Architecture
image::images/architecture.png[align="center",width=800]
```

How to add tables

- Place an anchor (e.g. [[my_anchor]]) above the table so that it can be referenced
- Configure to number and size of the columns (cols) and other options
- Always add a caption (preceded by dot on the example below)

example.adoc

```
[[tbl_capital_cities]]
[cols="4,6",options="header"]
.Capital cities
|===
| City Name | Country

| London | United Kingdom
| Madrid | Spain
| Ottawa| Canada
| Rome | Italy
|===

```

example.adoc Preview

Table 1. Capital cities

City Name	Country
London	United Kingdom
Madrid	Spain
Ottawa	Canada
Rome	Italy

Advanced tables

Span multiple columns by use of N+|, where N is the number of columns to span

Span multiple rows by use of .n+|, where .n is the number of rows to span

example.adoc

```
[[tbl_capital_cities]]
[cols="2,4,6",options="header"]
.Capital cities
|===
| Timezone | City Name | Country

3+| Europe
| UTC | London | United Kingdom

.2+| UTC+1
| Madrid | Spain
| Rome | Italy

3+| North America

| UTC -5 | Ottawa| Canada

|===
```

example.adoc Preview

Table 1. Capital cities

Timezone	City Name	Country
Europe		
UTC	London	United Kingdom
UTC+1	Madrid	Spain
	Rome	Italy
North America		
UTC -5	Ottawa	Canada

How to add code listings

Use the [source,format] directive where ‘format’ identifies the format of content

06-other-clauses.adoc

```
== Other clauses

This is an example listing in JavaScript Object Notat

[%unnumbered%]
[source,json]
----
{
  "type": "Feature",
  "geometry": {
    "type": "Point",
    "coordinates": [-2.682513,63.261372]
  },
  "properties": {
    "title": "SE1_OPER_SEA_GEC_1P_19780927T010430_197
  }
}

----
```

PUBLISHED

OGC COMMUNITY PRACTICE

6. OTHER CLAUSES

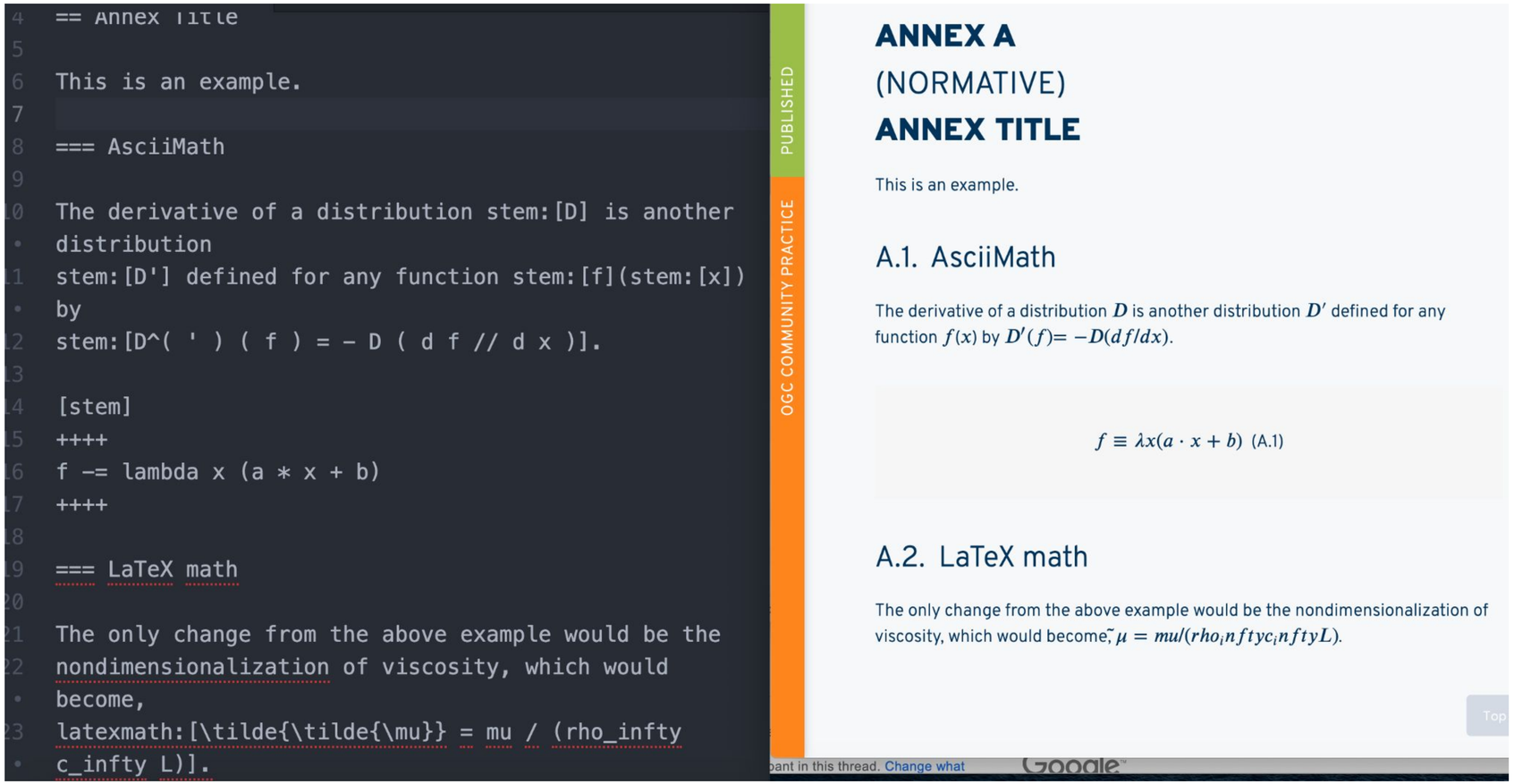
This is an example listing in JavaScript Object Notation (JSON).

```
{
        "type": "Feature",
  "geometry": {
    "type": "Point",
    "coordinates": [-2.682513,63.261372]
  },
        "properties": {
                "title": "SE1_OPER_SEA_GEC_1P_19780927T010430_19780927T0104
        }
}
```

NOTE: The [%unnumbered%] is optional. In this example it is used to prevent metanorma from numbering this particular code listing.

How to add mathematical formulae

- Metanorma AsciiDoc accepts mathematical input in: AsciiMath, LaTeX math, or MathML
- For simplicity, we recommend using AsciiMath



```
== Annex Title

This is an example.

=== AsciiMath

The derivative of a distribution stem:[D] is another
distribution
stem:[D'] defined for any function stem:[f](stem:[x])
by
stem:[D^( ' ) ( f ) = - D ( d f // d x )].

[stem]
++++
f -= lambda x (a * x + b)
++++

=== LaTeX math

The only change from the above example would be the
nondimensionalization of viscosity, which would
become,
latexmath:[\tilde{\tilde{\mu}} = mu / (rho_infty
c_infty L)].
```

PUBLISHED

OGC COMMUNITY PRACTICE

ANNEX A

(NORMATIVE)

ANNEX TITLE

This is an example.

A.1. AsciiMath

The derivative of a distribution D is another distribution D' defined for any function $f(x)$ by $D'(f) = -D(df/dx)$.

$$f \equiv \lambda x(a \cdot x + b) \quad \text{(A.1)}$$

A.2. LaTeX math

The only change from the above example would be the nondimensionalization of viscosity, which would become, $\tilde{\mu} = mu/(rho_i nfty c_i nfty L)$.

Top

ant in this thread. Change what

Google

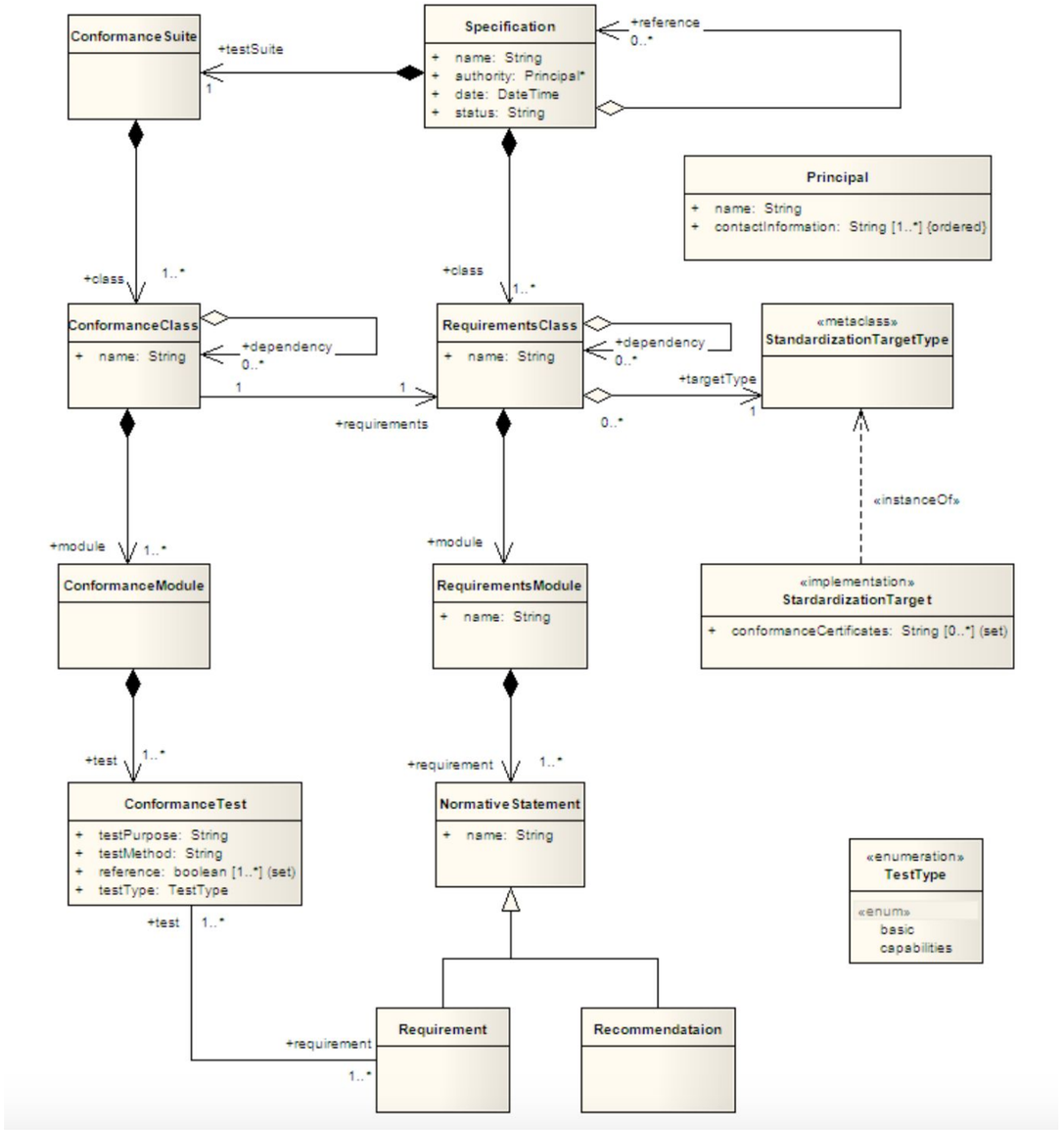
Source https://www.metanorma.org/author/topics/document-format/text/#mathematical-expressions

The ModSpec — An OGC Standard for Modular specifications

The ModSpec (OGC 08-131r3) lays requirements against other specifications by using a set-theoretic description of those specifications based on their structure as organized sets of criteria:

- Requirements classes
- Requirements
- Conformance classes
- Conformance tests

https://www.ogc.org/standards/modularspec



Documenting requirements classes in metanorma AsciiDoc

An example of a Requirements Class, documented using a definition list, is shown below.

```
[requirements_class]
====
[%metadata]
label:: /req/req-class-building
subject:: Implementation Specification
inherit:: <<rc_core,/req/req-class-core>>
inherit:: <<rc_construction,/req/req-class-construction>>
====
```

Requirements Class rendered

Requirements class 1	
/req/req-class-building	
Target type	Implementation Specification
Dependency	/req/req-class-core
Dependency	/req/req-class-construction

Documenting requirements in metanorma AsciiDoc

An example of a Requirement, documented using a definition list, is shown below.

```
[requirement]
====
[%metadata]
label:: /req/xsd-xml-rules/abc
part:: Data models faithful to the original UML model.
part:: Metadata models faithful to the original UML model.
description:: Logical models encoded as XSDs should be faithful
to the original
UML conceptual models.
====
```

Requirement rendered

Requirement 1	
/req/xsd-xml-rules/abc	
A	Data models faithful to the original UML model.
B	Metadata models faithful to the original UML model.
Logical models encoded as XSDs should be faithful to the original UML conceptual models.	

Notice that the parts are automatically numbered. This is a feature of metanorma.

Documenting conformance classes in metanorma AsciiDoc

An example of a Conformance Class, documented using a definition list, is shown below.

```
[conformance_class]
====
[%metadata]
label:: /conf/crs
subject:: <<rc_crs,Requirements Class 'Coordinate Reference
Systems by Reference'>>
inherit::
http://www.opengis.net/doc/IS/ogcapi-features-1/1.0#ats_core
classification:: Target Type:Web API
====
```

Conformance class rendered

Conformance class 1	
/conf/crs	
Requirements class	Requirements Class 'Coordinate Reference Systems by Reference'
Dependency	http://www.opengis.net/doc/IS/ogcapi-features-1/1.0#ats_core
Target type	Web API

Documenting conformance tests in metanorma AsciiDoc

An example of a Conformance Test (Abstract Test), documented using a definition list, is shown below.

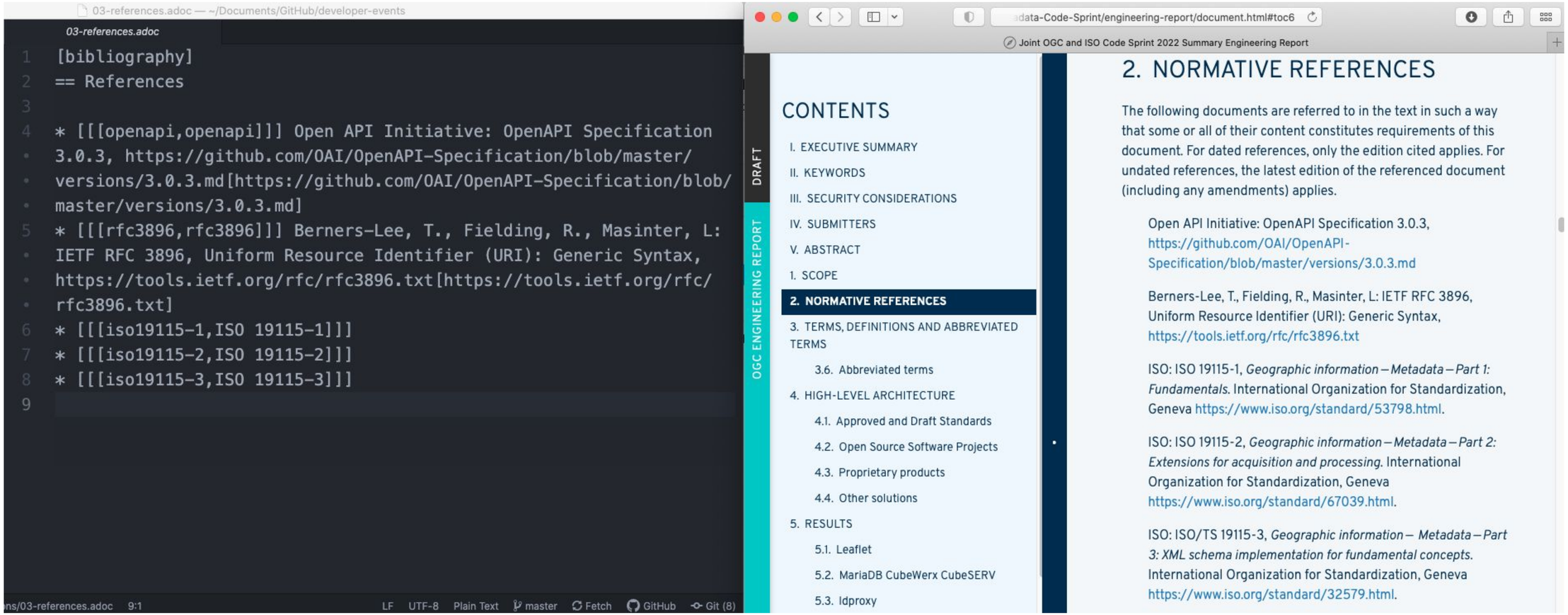
```
[abstract_test]
====
[%metadata]
label:: /conf/landing-page/root-op
subject:: <<req_landing-page_root-op,/req/landing-page/root-op>> +
<<req_landing-page_root-success,/req/landing-page/root-success>>
test-purpose:: Validate that a landing page can be retrieved from the expected location.
test-method::
+
--
. Issue an HTTP GET request to the URL {root}/
. Validate that a document was returned with a status code `200`
. Validate the contents of the returned document using test
<<ats_landing-page_root-success,/conf/landing-page/root-success>>.
--
====
```

Conformance Test rendered

Abstract test 1	
/conf/landing-page/root-op	
Requirement	/req/landing-page/root-op /req/landing-page/root-success
Test purpose	Validate that a landing page can be retrieved from the expected location.
Test method	1. Issue an HTTP GET request to the URL {root}/ 2. Validate that a document was returned with a status code 200 3. Validate the contents of the returned document using test /conf/landing-page/root-success.

Populating the References section

- Normative References pulled from a central register, using the relaton toolkit
- Saves editing time, and reduces the possibility of human error



Populating the Bibliography section

- Use the LNCS Style as per OGC Editorial Guidance
- Number the anchors of the bibliographic items using [[[anchor_id,serial_number]]]

az-bibliography.adoc

```
[appendix,obligation=informative]
[[annex-bibliography]]
[bibliography]
== Bibliography

* [[[ogc20-004,1]]], Vretanos, P.A, Kralidis, T., Heazel, C.: OGC
20-004: Draft OGC API - Records - Part 1: Core, http://docs.ogc.org/
DRAFTS/20-004.html

* [[[ogc21-045,2]]], Portele, C., Vretanos, P.A: OGC 21-045: OGC
Features and Geometries JSON - Part 1: Core, https://docs.ogc.org/
DRAFTS/21-045.html

* [[[stacref,3]]], STAC Community: SpatioTemporal Asset Catalog,
https://stacspec.org/en

* [[[Holmes2021,4]]], Holmes, C.: Static SpatioTemporal Asset
Catalogs in Depth, Available at https://medium.com/radiant-earth-
```


2022-09-26	0.1	G. Hobona	all	initial version

BIBLIOGRAPHY

[1] Vretanos, P.A, Kralidis, T., Heazel, C.: OGC 20-004: Draft OGC API—Records—Part 1: Core, http://docs.ogc.org/DRAFTS/20-004.html

[2] Portele, C., Vretanos, P.A: OGC 21-045: OGC Features and Geometries JSON—Part 1: Core, https://docs.ogc.org/DRAFTS/21-045.html

[3] STAC Community: SpatioTemporal Asset Catalog, https://stacspec.org/en

[4] Holmes, C.: Static SpatioTemporal Asset Catalogs in Depth, Available at https://medium.com/radiant-earth-insights/static-spatiotemporal-asset-catalogs-in-depth-710530934a84

Top

Thank You

Community

500+ International Members

110+ Member Meetings

60+ Alliance and Liaison partners

50+ Standards Working Groups

45+ Domain Working Groups

25+ Years of Not for Profit Work

10+ Regional and Country Forums

Innovation

120+ Innovation Initiatives

380+ Technical reports

Quarterly Tech Trends monitoring

Standards

65+ Adopted Standards

300+ products with 1000+ certified implementations

1,700,000+ Operational Data Sets

Using OGC Standards