OGC® DOCUMENT:

External identifier of this OGC® document: http://www.opengis.net/doc/spec/ogcapi-edr-3/1.0

Open Geospatial Consortium

OGC API-ENVIRONMENTAL DATA RETRIEVAL V1.2 PART 3 - NWS VIZ APPLICATION PROFILE

BEST PRACTICE
General

DRAFT

Version: 1.0
Submission Date:
Approval Date:
Publication Date:
Editor: Shane Mill (NOAA)

Notice: This document defines an OGC Best Practice on a particular technology or approach related to an OGC standard. This document is *not* an OGC Standard and may not be referred to as an OGC Standard. It is subject to change without notice. However, this document is an *official* position of the OGC membership on this particular technology topic.
Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

License Agreement

Use of this document is subject to the license agreement at https://www.ogc.org/license

Copyright notice

Copyright © 2025 Open Geospatial Consortium
To obtain additional rights of use, visithttps://www.ogc.org/legal

Note

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.

CONTENTS

I ABSTRACT

The aim of the NWSViz profile service profile is to provide a standard interface for accessing NWSViz profile data based on OGC API-EDR standard.

II KEYWORDS

The following are keywords to be used by search engines and document catalogues.

ogcdoc, OGC document, API, openapi, html, profile

PREFACE

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.

IV SECURITY CONSIDERATIONS

No security considerations have been made for this Service Profile.

V SUBMITTERS

All questions regarding this submission should be directed to the editor or the submitters:

Shane Mill	**NOAA**

VI CONTRIBUTORS

Additional contributors to this Profile include the following:

Individual name(s), Organization

PREFACE

PREFACE

NOTE: 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.

1

SCOPE

1 SCOPE

NOTE: This document defines the NWSViz profile Profile of the OGC API-EDR Part 1: Core Standard.

2

CONFORMANCE

2 CONFORMANCE

Conformance to the NWSViz profile Profile (this document) can be tested by inspection. The test suite is provided in Annex A.

This Standard contains normative language and thus places requirements on conformance, or mechanism for adoption, of candidate standards to which this Standard applies. In particular:

- OGC API-EDR Requirements Class: Core specifies the core requirements which shall be met by all standards claiming conformance to this Standard.

3

NORMATIVE REFERENCES

3 NORMATIVE REFERENCES

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO: ISO 19106, *Geographic information — Profiles*. International Organization for Standardization, Geneva https://www.iso.org/standard/26011.html.

Mark Burgoyne, Dave Blodgett, Chuck Heazel, Chris Little: OGC 19-086r4, *OGC API — Environmental Data Retrieval Standard*. Open Geospatial Consortium (2021). http://www.opengis.net/doc/IS/ogcapi-edr-1/1.0.0.

https://docs.ogc.org/is/17-069r4/17-069r4.html, OGC APIFeatures — Part 1: Core, Open Geospatial Consortium (2022).

https://docs.ogc.org/is/19-072/19-072.html, OGC API — Common — Part 1: Core, Open Geospatial Consortium (2023).

http://docs.ogc.org/DRAFTS/20-024.html, OGC API — Common — Part 2: Geospatial Data (Draft), Open Geospatial Consortium

Policy SWG: OGC 08-131r3, *The Specification Model — Standard for Modular specifications*. Open Geospatial Consortium (2009). https://portal.ogc.org/files/?artifact_id=34762&version=2.

OpenAPI Initiative (OAI). **OpenAPI Specification 3.0** [online]. 2024 [viewed 2025-01-03]. The latest patch version at the time of publication of this standard was 3.0.4, available at https://spec.openapis.org/oas/v3.0.4

OpenAPI Initiative (OAI). **OpenAPI Specification 3.1** [online]. 2024 [viewed 2025-01-03]. The latest patch version at the time of publication of this standard was 3.1.1, available at https://spec.openapis.org/oas/v3.1.1

4

TERMS AND DEFINITIONS

4 TERMS AND DEFINITIONS

This document uses the terms defined in OGC Policy Directive 49, which is based on the ISO/IEC Directives, Part 2, Rules for the structure and drafting of International Standards. In particular, the word “shall” (not “must”) is the verb form used to indicate a requirement to be strictly followed to conform to this document and OGC documents do not use the equivalent phrases in the ISO/IEC Directives, Part 2.

This document also uses terms defined in the OGC Standard for Modular specifications (OGC 08-131r3), also known as the ‘ModSpec’. The definitions of terms such as standard, specification, requirement, and conformance test are provided in the ModSpec.

For the purposes of this document, the following additional terms and definitions apply.

4.1. Collection

Body of resources that belong or are used together. An aggregate, set, or group of related resources.

[**SOURCE:** OGC 20-024]

4.2. Conformance Module; Conformance Test Module

A set of related conformance classes and their associated components.

Note 1 to entry: When no ambiguity is possible, the word `test` may be omitted. i.e. conformance test module is the same as conformance module. Conformance modules may be nested in a hierarchical way.

[**SOURCE:** OGC 08-131r5]

4.3. Conformance Class; Conformance Test Class

A set of conformance tests that must be passed to receive a single certificate of conformance.

Note 1 to entry: When no ambiguity is possible, the word *test* may be left out, so conformance test class maybe called a conformance class.

[**SOURCE:** OGC 08-131r5]

4.4. Conformance Test

A test, abstract or real, of one or more requirements contained within a standard, or set of standards.

[**SOURCE:** OGC 08-131r5]

4.5. Requirement

Expression in the content of a standard conveying criteria to be fulfilled if compliance with the standard is to be claimed and from which no deviation is permitted.

[**SOURCE:** OGC 08-131r5]

4.6. Requirements Class

An aggregate of requirements with a single standardization target type that must all be satisfied to pass a conformance test Class.

[**SOURCE:** OGC 08-131r5]

4.7. Requirements Module

A set of related requirement classes and their associated components.

[**SOURCE:** OGC 08-131r5]

4.8. Standardization Goal

A concise statement of the problem that the standard helps address and the strategy envisioned for achieving a solution. This strategy typically identifies real-world entities that need to be modified or constrained. At the abstract level, those entities are the Standardization Target Types.

[**SOURCE:** OGC 08-131r5]

4.9. Standardization Target

Entity to which some requirements of a standard apply.

Note 1 to entry: The standardization target is the entity which may receive a certificate of conformance for a requirements class.

[**SOURCE:** OGC 08-131r5]

4.10. Standardization Target Type

Type of entity or set of entities to which the requirements of a standard apply

Note 1 to entry: For example, the standardization target type for The OGC API – Features Standard are Web APIs. The standardization target type for the CDB Standard is “datastore”. It is important to understand that a standard’s root standardization target type can have sub-types, and that there can be a hierarchy of target types. For example, a Web API can have sub types of client, server, security, and so forth. As such, each requirements class can have a standardization target type that is a sub-type of the root.

[**SOURCE:** OGC 08-131r5]

5 CONVENTIONS

5 CONVENTIONS

This sections provides details and examples for any conventions used in the document. Examples of conventions are symbols, abbreviations, use of XML schema, or special notes regarding how to read the document.

5.1. Identifiers

The normative provisions in this standard are denoted by the URI

`http://www.opengis.net/doc/spec/ogcapi-edr-3/1.0`

All requirements and conformance tests that appear in this document are denoted by partial URIs which are relative to this base.

5.1.1. Shortcuts

In the interest of readability, the following terms will be used as shorthand for more complex text:

- Profile: A Profile is a standard or specification which restricts and/or extends an existing standard. This standard defines the rules for creating a profile of the OGC API-Environmental Data Retrieval Standard. The term “Profile” will be used in this document as shorthand for “profile of the OGC API-Environmental Data Retrieval Standard”.
- OGC API-EDR: The term OGC API-EDR will be used in this document as shorthand for the term “OGC API-Environmental Data Retrieval Standard”

6

CONTEXT

6 CONTEXT

6.1. Standardization Goal

The goal of this profile is to ensure interoperabilty between NWSViz profile data implementations of the OGC API-Environmental Data Retrieval Standard (OGC API-EDR).

The OGC API-EDR Standard does not try to address every possible application domain. Rather, it provides a foundation which can be tailored for a specific domain. The result of this tailoring is a domain specific “profile” of the OGC API-EDR Standard.

7

REQUIREMENTS CLASS CORE

7 REQUIREMENTS CLASS CORE

REQUIREMENTS CLASS 1: REQUIREMENTS CLASS 'NWSVIZ APPLICATION'	
IDENTIFIER	http://www.example.org/1.0/req/req-class-nwsviz-application
CONFORMANCE CLASS	Conformance class A.1: http://www.example.org/1.0/conf-class-nwsviz-application
TARGET-TYPE	NWSViz Application Profile Standard
NORMATIVE STATEMENTS	Requirement 1: /req/nwsviz-application/edr-conformant Requirement 1-3: /req/nwsviz-application/parameter-names Requirement 2: /req/nwsviz-application_data/root Requirement 3: /req/nwsviz-application/collectionid Requirement 4: /req/nwsviz-application/extent Requirement 5: /req/nwsviz-application/NBM-parameter-names Requirement 6: /req/nwsviz-application/MRMS-parameter-names Requirement 7: /req/nwsviz-application/data-query Requirement 8: /req/nwsviz-application/output-format Requirement 9: /req/nwsviz-application/data-query-position Requirement 10: /req/nwsviz-application/data-query-items Requirement 11: /req/nwsviz-application/data-query-locations Requirement 12: /req/nwsviz-application/data-query-location Requirement 13: /req/nwsviz-application/data-query-instances Requirement 14: /req/nwsviz-application/status-codes Requirement 15: /req/nwsviz-application/msg-bodies Requirement 16: /req/nwsviz-application/links

7.1. Profiling Requirements

Implementations of the Profile are conformant with OGC API-EDR Part 1

REQUIREMENT 1	
IDENTIFIER	/req/nwsviz-application/edr-conformant
INCLUDED IN	Requirements class 1: http://www.example.org/1.0/req/req-class-nwsviz-application

REQUIREMENT 1	
STATEMENT	The service *SHALL* be compliant with OGC API-Environmental Data Retrieval core v1.2.

7.2. Platform Resources

OGC API — Common defines a set of common capabilities which are applicable to any OGC Web API. Those capabilities provide the platform upon which resource-specific APIs can be built. This section describes those capabilities and any modifications needed to better support spatio-temporal data resources.

Table 1 — Platform Resource Paths

PATH TEMPLATE	METHOD	RESOURCE
{root}/	GET	Landing page
{root}/api	GET	API Description (optional)
{root}/conformance	GET	Conformance Classes

Where: {root} = Base URI for the API server

7.2.1. API Landing Page

Path = {root}/

Dependencies

- OGC API — Common — Part 1: Core
- OGC API-Environmental Data Retrieval Standard — Part 1.2: Core
- OGC API-Environmental Data Retrieval Standard — Part 3 Service Profiles

The landing page provides links that support exploration of the resources offered via the API. The most important component of a landing page is a list of links. The Landing Page resource is initially defined in the Core conformance class of the OGC API — Common — Part 1 Standard. The OGC API-Environmental Data Retrieval Standard Standard does not make any changes to this definition.

The normative JSON Schema for an OGC API-EDR Landing Page is defined in the LandingPage.yaml document. While this schema provides a rich body of information about the API, only the Links property is required.

Profiles of the OGC API-Environmental Data Retrieval Standard are expected to provide a richer description of the API. The additional content that Profiles should mandate is defined in the following requirements.

REQUIREMENT 2	
IDENTIFIER	`/req/nwsviz-application_data/root`
INCLUDED IN	Requirements class 1: `http://www.example.org/1.0/req/req-class-nwsviz-application`
STATEMENT	The landing page *SHALL* have the following infomation
A	The **Title** property value *SHALL* be "OGC API — Environmental Data Retrieval for Dynamic Ensemble Scenarios for IDSS"
B	The **Description** property value *SHALL* be "Implementation of Pygeoapi for OGC API — EDR to support DESI"
C	The **Links** property *SHALL* define the links that SHALL be included in the Root response and *SHALL* be populated with href and rel properties.
D	The **provider** property *SHALL* be included in the Root response and *SHALL* be populated with name and url properties.
E	The **contact** property *SHALL* be included in the Root response and *SHALL* be populated with an email property
F	The **Links** property *SHALL* include the following link to the OpenAPI definition of the profile:
STATEMENT	`{ "title": "OpenAPI definition of NWSViz Application profile", "href": "https://www.example.org/edr/profile/nwsviz/openapi.json", "rel": "profile", "type": "application/json" }`

7.2.2. API Definition

Path = {root}/api

Dependencies

- OGC API — Common — Part 1: Core
- OGC API-Environmental Data Retrieval Standard

Every API is required to provide a definition document that describes the capabilities of that API. This definition document can be used by developers to understand the API, by software clients to connect to the server, or by development tools to support the implementation of servers and clients. The API Definition resource is initially defined in the Core conformance class of the OGC API — Common — Part 1 Standard. The OGC API-Environmental Data Retrieval Standard does not make any changes to this definition.

NOTE: At this time only OpenAPI 3.0 and OpenAPI 3.1 documents are supported by OGC Web API Standards.

Profiles of the OGC API-Environmental Data Retrieval Standard are required to provide an OpenAPI 3.1 document. This document extends the API definition provided by the OGC API-EDR Standard. These extensions reflect the additional requirements added by the Profile. Implementors of the profile will then build on that document to produce the API definition document for their implementation.

7.2.3. Declaration of Conformance Classes

Path = {root}/conformance

Dependencies

- OGC API — Common — Part 1: Core
- OGC API-Environmental Data Retrieval Standard
- OGC API-Environmental Data Retrieval Standard — Part 3 Service Profiles

To support “generic” clients that want to access implementations of multiple OGC API Standards and extensions — and not “just” a specific API server, the API has to declare the conformance classes it claims to have implemented. The Conformance Classes resource is initially defined in the Core conformance class of the OGC API — Common — Part 1 Standard. The OGC API-Environmental Data Retrieval Standard Standard does not make any changes to this definition.

Profiles of the OGC API-Environmental Data Retrieval Standard have additional requirements governing which Conformance Classes and identifiers must be included in this resource.

NOTE 1: OpenAPI 3.0 and OpenAPI 3.1 are two distinct Conformance Classes in the OGC API-EDR Standard. This requirement can be addressed in a Profile by including the appropriate conformance classes at {root}/conformance.

NOTE 2: Get guidence from the OGC Naming Authority on valid URIs for Profiles.

7.3. Spatio-temporal and Information Resources

Table 2 — Spatial-temporal and Information Resource Paths

PATH TEMPLATE	METHOD	RESOURCE
{root}/collections	GET	Metadata describing the Collections of data available from this API.
{root}/collections/{collectionId}	GET	Metadata describing the Collection of data which has the unique identifier {collectionId}

Where:

- {root} = Base URI for the API server
- {collectionId} = an identifier for a specific Collection of data

7.3.1. Collections

OGC API implementations typically organize their geospatial resources into Collections. Information about those is accessed through the /collections path and the http://www.opengis.net/def/rel/ogc/1.0/data link relation.

Path = {root}/collections

Dependencies

- OGC API — Common — Part 2: Geospatial Data
- OGC API-Environmental Data Retrieval Standard

The Collections resource is initially defined in the `Collections` conformance class of the OGC API — Common — Part 2 Standard. The OGC API-Environmental Data Retrieval Standard Standard does not make any changes to this definition.

7.3.2. Collection Description

Each resource Collection is described by a set of metadata. That metadata can be accessed directly using the /collections/{collectionId} path and as an entry in the Collections property of the /collections response.

Path:

- {root}/collections (returns metadata for every Collection)
- {root}/collections/{collectionId} (returns metadata for the specified Collection)

Dependencies

- OGC API — Common — Part 2: Geospatial Data
- OGC API-Environmental Data Retrieval Standard

7.3.2.1. Collection ID restictions

REQUIREMENT 3	
IDENTIFIER	`/req/nwsviz-application/collectionid`
INCLUDED IN	Requirements class 1: `http://www.example.org/1.0/req/req-class-nwsviz-application`
STATEMENT	A Collection id should identify the source and type of NWP model
A	The Collection Ids *SHALL* contain the following attributes

STATEMENT	id	description
	NBM_icechunk	National Blend of Models
	MRMS_icechunk	Multi-Radar/Multi-Sensor Radar Products

7.3.2.2. Extent property restrictions

The Collection metadata includes an Extent property which defines a spatial-temporal envelope that encompasses the geospatial data in the Collection.

REQUIREMENT 4	
IDENTIFIER	`/req/nwsviz-application/extent`
INCLUDED IN	Requirements class 1: `http://www.example.org/1.0/req/req-class-nwsviz-application`
STATEMENT	The Collection *SHALL* have minimum spatial bounds
A	The Profile *SHALL* have a spatial extent and contain an additional attribute named “locations” containing the a list of the available locations within the collection.
B	The Profile *SHALL* have a temporal extent containing an interval and values.

REQUIREMENT 4	
C	The Profile *SHALL* contain an attribute named instances which contains a list of available instances for the collection.
D	The Profiles *SHALL* contain a custom query dimension with the name “ensemble” containing values and ids.
E	The Profile *SHOULD* have a vertical extent if applicable

7.3.2.3. Parameter_names property restrictions

The Collection metadata includes an Parameter_names property which defines the data parameters that are available in the Collection

REQUIREMENT 5	
IDENTIFIER	`/req/nwsviz-application/NBM-parameter-names`
INCLUDED IN	Requirements class 1: `http://www.example.org/1.0/req/req-class-nwsviz-application`
STATEMENT	The NBM Collection *SHALL* have a defined dictionary of parameter_names
A	A NBM Collection *SHALL* follow the following schema where the metadata is dynamically loaded from the underlying store.

STATEMENT	"parameter-names":{
	"apparent_temperature":{
	"Type":"Parameter",
	"id":"apparent_temperature",
	"description":"apparent_temperature",
	"unit":{
	"symbol":"K"
	},
	"temporal":{
	"values":[
	"2025-09-22T01:00:00",
	"2025-09-22T02:00:00",
	"...etc"
	]
	},
	"attrs":{
	"grib_section3":[
	0,
	3744965,
	0,
	0,
	30,
	1,
	0,
	6371200,
	255,
	255,
	255,
	255,
	2345,

REQUIREMENT 5


```
                    1597,
                    19229000,
                    233723400,
                    48,
                    25000000,
                    265000000,
                    2539703,
                    2539703,
                    0,
                    80,
                    25000000,
                    25000000,
                    -90000000,
                    0
                ],
                "long_name":"Apparent Temperature",
                "short_name":"APTMP",
                "units":"K",
                "originating_center":"US National Weather Service - NCEP (WMC)",
                "originating_sub_center":"NWS Meteorological Development
        Laboratory",
                "master_table_info":"Version Implemented on 7 November 2001",
                "product_definition_template_number":"Analysis or forecast at a
        horizontal level or in a horizontal layer at a point in time. (see Template
        4.0)",
                "type_of_generating_process":"Forecast",
                "type_of_first_fixed_surface":"Specified Height Level Above Ground
        (m)",
                "type_of_second_fixed_surface":"Missing (unknown)",
                "crs_wkt":"PROJCRS[\"unknown\",BASEGEOGCRS[\"unknown\",
        DATUM[\"unknown\",ELLIPSOID[\"unknown\",6371200,0,LENGTHUNIT[\"metre\",1,
        ID[\"EPSG\",9001]]]],PRIMEM[\"Greenwich\",0,ANGLEUNIT[\"degree\",0.01745
        32925199433],ID[\"EPSG\",8901]]],CONVERSION[\"unknown\",METHOD[\"Lambert
        Conic Conformal (2SP)\",ID[\"EPSG\",9802]],PARAMETER[\"Latitude of false
        origin\",25,ANGLEUNIT[\"degree\",0.0174532925199433],ID[\"EPSG\",8821]],
        PARAMETER[\"Longitude of false origin\",265,ANGLEUNIT[\"degree\",0.017
        4532925199433],ID[\"EPSG\",8822]],PARAMETER[\"Latitude of 1st standard
        parallel\",25,ANGLEUNIT[\"degree\",0.0174532925199433],ID[\"EPSG\",8823]],
        PARAMETER[\"Latitude of 2nd standard parallel\",25,ANGLEUNIT[\"degree\",0.0
        174532925199433],ID[\"EPSG\",8824]],PARAMETER[\"Easting at false origin\",0,
        LENGTHUNIT[\"metre\",1],ID[\"EPSG\",8826]],PARAMETER[\"Northing at false
        origin\",0,LENGTHUNIT[\"metre\",1],ID[\"EPSG\",8827]]],CS[Cartesian,2],
        AXIS[\"(E)\",east,ORDER[1],LENGTHUNIT[\"metre\",1,ID[\"EPSG\",9001]]],
        AXIS[\"(N)\",north,ORDER[2],LENGTHUNIT[\"metre\",1,ID[\"EPSG\",9001]]]]",
                "gridlength_x_direction":2539.703,
                "gridlength_y_direction":2539.703,
                "latitude_first_gridpoint":19.229,
                "longitude_first_gridpoint":233.7234,
                "standard_name":"apparent_air_temperature",
                "coordinates":"forecast_reference_time lead_time specified_height_
        level_above_ground",
                "_FillValue":"AAAAAAAA+H8="
            }
        }
    }
```


REQUIREMENT 6

IDENTIFIER /req/nwsviz-application/MRMS-parameter-names

REQUIREMENT 6	
INCLUDED IN	Requirements class 1: `http://www.example.org/1.0/req/req-class-nwsviz-application`
STATEMENT	The MRMS Collection *SHALL* have a defined dictionary of parameter_names
A	A MRMS Collection *SHALL* follow the following schema where the metadata is dynamically loaded from the underlying store.

STATEMENT	{
	"parameter-names":{
	"EchoTop_18":{
	"Type":"Parameter",
	"id":"EchoTop_18",
	"description":"EchoTop_18",
	"temporal":{
	"values":[
	"2025-10-01T18:24:39",
	"2025-10-01T18:32:41",
	"...etc"
	]
	},
	"attrs":{
	"_FillValue":"AAAAAAAA+H8="
	}
	},
	"EchoTop_30":{
	"Type":"Parameter",
	"id":"EchoTop_30",
	"description":"EchoTop_30",
	"temporal":{
	"values":[
	"2025-10-01T17:58:37",
	"2025-10-01T18:02:40",
	"...etc"
	]
	},
	"attrs":{
	"_FillValue":"AAAAAAAA+H8="
	}
	},
	"EchoTop_60":{
	"Type":"Parameter",
	"id":"EchoTop_60",
	"description":"EchoTop_60",
	"temporal":{
	"values":[
	"2025-10-01T18:06:38",
	"2025-10-01T18:10:35",
	"...etc"
	]
	},
	"attrs":{
	"_FillValue":"AAAAAAAA+H8="
	}
	},
	"H50Above0C":{
	"Type":"Parameter",
	"id":"H50Above0C",
	"description":"H50Above0C",
	"temporal":{
	"values":[
	"2025-10-01T18:00:34",
	"2025-10-01T18:04:39",
	"...etc"
	]

```
            },
            "attrs":{
               "_FillValue":"AAAAAAAA+H8="
            }
         },
         "H60AboveM20C":{
            "Type":"Parameter",
            "id":"H60AboveM20C",
            "description":"H60AboveM20C",
            "temporal":{
               "values":[
                  "2025-10-01T18:26:38",
                  "2025-10-01T18:28:40",
                  "...etc"
               ]
            },
            "attrs":{
               "_FillValue":"AAAAAAAA+H8="
            }
         },
         "LightningProbabilityNext30minGrid":{
            "Type":"Parameter",
            "id":"LightningProbabilityNext30minGrid",
            "description":"LightningProbabilityNext30minGrid",
            "temporal":{
               "values":[
                  "2025-10-01T17:58:37",
                  "2025-10-01T18:00:34",
                  "2025-10-01T18:02:35",
                  "...etc"
               ]
            },
            "attrs":{
               "_FillValue":"AAAAAAAA+H8="
            }
         },
         "POSH":{
            "Type":"Parameter",
            "id":"POSH",
            "description":"POSH",
            "temporal":{
               "values":[
                  "2025-10-01T18:12:36",
                  "2025-10-01T18:16:39",
                  "...etc"
               ]
            },
            "attrs":{
               "_FillValue":"AAAAAAAA+H8="
            }
         },
         "PrecipFlag":{
            "Type":"Parameter",
            "id":"PrecipFlag",
            "description":"PrecipFlag",
            "temporal":{
               "values":[
                  "2025-10-01T18:04:00",
                  "2025-10-01T18:06:00",
                  "...etc"
               ]
            },
            "attrs":{
               "_FillValue":"AAAAAAAA+H8="
            }
         },
         "RadarOnly_QPE_01H":{
```

```
            "Type":"Parameter",
            "id":"RadarOnly_QPE_01H",
            "description":"RadarOnly_QPE_01H",
            "temporal":{
               "values":[
                  "2025-10-01T17:58:00",
                  "2025-10-01T18:00:00",
                  "...etc"
               ]
            },
            "attrs":{
               "_FillValue":"AAAAAAAA+H8="
            }
         },
         "RadarOnly_QPE_03H":{
            "Type":"Parameter",
            "id":"RadarOnly_QPE_03H",
            "description":"RadarOnly_QPE_03H",
            "temporal":{
               "values":[
                  "2025-10-01T18:00:00"
               ]
            },
            "attrs":{
               "_FillValue":"AAAAAAAA+H8="
            }
         },
         "RadarOnly_QPE_15M":{
            "Type":"Parameter",
            "id":"RadarOnly_QPE_15M",
            "description":"RadarOnly_QPE_15M",
            "temporal":{
               "values":[
                  "2025-10-01T18:00:00",
                  "2025-10-01T18:15:00",
                  "...etc"
               ]
            },
            "attrs":{
               "_FillValue":"AAAAAAAA+H8="
            }
         },
         "RadarOnly_QPE_72H":{
            "Type":"Parameter",
            "id":"RadarOnly_QPE_72H",
            "description":"RadarOnly_QPE_72H",
            "temporal":{
               "values":[
                  "2025-10-01T18:00:00"
               ]
            },
            "attrs":{
               "_FillValue":"AAAAAAAA+H8="
            }
         },
         "Reflectivity_-10C":{
            "Type":"Parameter",
            "id":"Reflectivity_-10C",
            "description":"Reflectivity_-10C",
            "temporal":{
               "values":[
                  "2025-10-01T18:02:35",
                  "2025-10-01T18:06:38",
                  "...etc"
               ]
            },
            "attrs":{
```

REQUIREMENT 6


```
                        "_FillValue":"AAAAAAAA+H8="
                     }
                  },
                  "Reflectivity_-20C":{
                     "Type":"Parameter",
                     "id":"Reflectivity_-20C",
                     "description":"Reflectivity_-20C",
                     "temporal":{
                        "values":[
                           "2025-10-01T18:20:37",
                           "2025-10-01T18:22:35",
                           "...etc"
                        ]
                     },
                     "attrs":{
                        "_FillValue":"AAAAAAAA+H8="
                     }
                  },
                  "Reflectivity_0C":{
                     "Type":"Parameter",
                     "id":"Reflectivity_0C",
                     "description":"Reflectivity_0C",
                     "temporal":{
                        "values":[
                           "2025-10-01T17:58:37",
                           "2025-10-01T18:00:34",
                           "...etc"
                        ]
                     },
                     "attrs":{
                        "_FillValue":"AAAAAAAA+H8="
                     }
                  },
                  "VII":{
                     "Type":"Parameter",
                     "id":"VII",
                     "description":"VII",
                     "temporal":{
                        "values":[
                           "2025-10-01T18:00:34",
                           "2025-10-01T18:02:40",
                           "...etc"
                        ]
                     },
                     "attrs":{
                        "_FillValue":"AAAAAAAA+H8="
                     }
                  },
                  "VIL":{
                     "Type":"Parameter",
                     "id":"VIL",
                     "description":"VIL",
                     "temporal":{
                        "values":[
                           "2025-10-01T18:00:34",
                           "2025-10-01T18:04:39",
                           "...etc"
                        ]
                     },
                     "attrs":{
                        "_FillValue":"AAAAAAAA+H8="
                     }
                  }
               }
            }
```

7.4. Query Resources

Table 4 — Query Resource Paths

PATH TEMPLATE	METHOD	RESOURCE
{root}/collections/{collectionId}/{queryType}	GET, POST (Optional)	Retrieve data according to the query pattern from a Collection with the unique identifier {collectionId}
{root}/collections/{collectionId}/instances	GET	Retrieve metadata about instances of a collection
{root}/collections/{collectionId}/instances/{instanceId}	GET	Retrieve metadata from a specific instanceof a Collection with the unique identifiers{collectionId} and {instanceId}
{root}/collections/{collectionId}/instances/{instanceId}/{queryType}	GET, POST (Optional)	Retrieve data according to the query pattern from a specific instance of a Collection with the unique identifiers{collectionId} and {instanceId}

Where:

- {root} = Base URI for the API server
- {collectionId} = an identifier for a specific Collection of data
- {instanceId} = an identifier for a specific version or instance of a Collection of data
- {queryType} = an identifier for a specific query pattern to retrieve data from a specific Collection of data

Path = {root}/collections/{collectionId}/{queryType}

Dependencies: OGC API-Environmental Data Retrieval Standard

REQUIREMENT 7	
IDENTIFIER	`/req/nwsviz-application/data-query`
INCLUDED IN	Requirements class 1: `http://www.example.org/1.0/req/req-class-nwsviz-application`
STATEMENT	The collections *SHALL* support a defined set of data queries.
A	The collections *SHALL* support the following data queries: • Position • Locations

REQUIREMENT 7
• Items • Instances

7.4.1. Parameters

The following parameters are supported by all OGC API-EDR queries.

7.4.1.1. Output Format parameter

Data format for the output data (available options are listed in the collectionsresponse).

REQUIREMENT 8	
IDENTIFIER	`/req/nwsviz-application/output-format`
INCLUDED IN	Requirements class 1: `http://www.example.org/1.0/req/req-class-nwsviz-application`
STATEMENT	Collections *SHALL* support the following formats:
A	A format with the label **CoverageJSON** *SHALL* provide CoverageJSON output which is described by https://docs.ogc.org/cs/21-069r2/21-069r2.html
B	A format with the label **NetCDF4** *SHALL* provide CF-NetCDF output which is described by https://cfconventions.org/Data/cf-conventions/cf-conventions-1.10/ cf-conventions.html
C	A format with the label **Zarr** *SHALL* provide Zarr output which is described by https://zarr-specs.readthedocs.io/en/latest/v3/core/index.html

7.4.1.2. Parameter queryType

Path — Instance Query {root}/collections/{collectionId}/instances/{instanceId}/{queryType}

7.4.2. Position Query

The Position query returns data for the requested coordinate. Logic for identifying the best match for the coordinate will depend on the Collection and is at the discretion of the query service implementer.

Path = {root}/collections/{collectionId}/instances/{instanceId}/position

Dependencies: OGC API-Environmental Data Retrieval Standard

REQUIREMENT 9	
IDENTIFIER	`/req/nwsviz-application/data-query-position`
INCLUDED IN	Requirements class 1: `http://www.example.org/1.0/req/req-class-nwsviz-application`
STATEMENT	All Collections in the Service *SHALL* provide support for **Position** queries
A	The **Position** query *SHALL* support the following output formats: • CoverageJSON
B	The default output format for the **Position** query *SHALL* be **CoverageJSON**
C	The **Position** query *SHALL* support the following HTTP methods: • GET

7.4.3. Items Query

The Items query allows a user to query a resource based on a unique identifier.

Path = {root}/collections/{collectionId}/instances/{instanceId}/items

Dependencies: OGC API-Environmental Data Retrieval Standard

REQUIREMENT 10		
IDENTIFIER	`/req/nwsviz-application/data-query-items`	
INCLUDED IN	Requirements class 1: `http://www.example.org/1.0/req/req-class-nwsviz-application`	
STATEMENT	All Collections in the Service *SHALL* provide support for **Items** queries	
A	The **Items** query *SHALL* return a connection to a Virtual Zarr Store that can be accessed via Xarray. For example:	

	`import xarray as xr`
	
	`ds=xr.open_zarr('link:++https://edr-api-desi-c.mdl.nws.noaa.gov/collections/MRMS_icechunk/instances/2025-10-01T18:00:00/items/zarr/EchoTop_30/0++[]')`
	`# Where EchoTop_30 is the element name and 0 is the zoom level`	

STATEMENT	part	The **Items** query *SHALL* support the following HTTP methods: • GET
	part	The **Items** query *SHALL* have an item identifier that follows the following syntax:

REQUIREMENT 10	
	• zarr/<element>/<zoom>/<crs>/<unit>

7.4.4. Locations Query

The Location query returns a GeoJSON Feature Collection containing the available locations with a collection

Path = {root}/collections/{collectionId}/instances/{instanceId}/locations

Dependencies: OGC API-Environmental Data Retrieval Standard

REQUIREMENT 11	
IDENTIFIER	`/req/nwsviz-application/data-query-locations`
INCLUDED IN	Requirements class 1: `http://www.example.org/1.0/req/req-class-nwsviz-application`
STATEMENT	All Collections in the Service *SHALL* provide support for **Locations** queries
A	The **Locations** query *SHALL* support the following output formats: • GeoJSON
B	The default output format for the **Locations** query *SHALL* be **GeoJSON**
C	The **Locations** query *SHALL* support the following HTTP methods: • GET • POST
D	The **Locations** query *SHALL* provide a listing of the available locations in a collection that conforms to GeoJSON. The GeoJSON should contain properties as shown below. Note that the example below only contains one parameter in the parameter-names-dict, but in reality it should contain all of the parameters listed in the parameter-names property.

STATEMENT	<pre>{
	"type":"FeatureCollection",
	"features":[
	{
	"type":"Feature",
	"geometry":{
	"type":"Polygon",
	"coordinates":[
	[
	[
	2681912.2261628294,
	-263793.73346456443
	],
	[
	2681912.2261628294,
	3789572.2545354357
	],</pre>

REQUIREMENT 11


```
                        [
                           -3271151.605837171,
                           3789572.2545354357
                        ],
                        [
                           -3271151.605837171,
                           -263793.73346456443
                        ],
                        [
                           2681912.2261628294,
                           -263793.73346456443
                        ]
                     ]
                  ]
               },
               "properties":{
                  "instance":"2025-09-22T00:00:00",
                  "name":"conus",
                  "name_alt":"link:++http://edr-api-desi-c.mdl.nws.noaa.gov/
collections/NBM_icechunk/instances/2025-09-22T00:00:00/locations/conus?f=
json++[]",
                  "edrqueryendpoint":"link:++http://edr-api-desi-c.mdl.nws.noaa.
gov/collections/NBM_icechunk/instances/2025-09-22T00:00:00/locations/conus?
f=json++[]",
                  "parameter-names":[
                     "apparent_temperature",
                     "ceiling",
                     "ceiling_probability",
                     "cloud_base",
                     "conditional_probability_of_precipitation_type",
                     "convective_available_potential_energy",
                     "convective_available_potential_energy_percentiles",
                     "convective_available_potential_energy_standard_deviation",
                     "dewpoint",
                     "dewpoint_standard_deviation",
                     "downward_shortwave_radiation_flux",
                     "dry_thunderstorm_probability",
                     "echo_top",
                     "ellrod_index",
                     "fosberg_index_06_hour",
                     "ice_accumulation_01_hour",
                     "ice_accumulation_06_hour",
                     "ice_accumulation_06_hour_percentiles",
                     "low_level_turbulence",
                     "low_level_wind_shear_altitude",
                     "low_level_wind_shear_direction",
                     "low_level_wind_shear_magnitude",
                     "maximum_reflectivity",
                     "maximum_relative_humidity_12_hour",
                     "maximum_temperature_12_hour",
                     "maximum_temperature_12_hour_standard_deviation",
                     "minimum_relative_humidity_12_hour",
                     "minimum_temperature_12_hour",
                     "minimum_temperature_12_hour_standard_deviation",
                     "mixing_height",
                     "precipitable_water",
                     "precipitable_water_percentiles",
                     "precipitation_01_hour_probability",
                     "precipitation_06_hour_probability",
                     "precipitation_12_hour_probability",
                     "precipitation_accumulation_01_hour",
                     "precipitation_accumulation_06_hour",
                     "precipitation_duration_12_hour",
                     "predominant_weather",
                     "relative_humidity",
                     "sea_surface_temperature",
                     "snow_accumulation_01_hour",
```

```
                        "snow_accumulation_01_hour_percentiles",
                        "snow_accumulation_06_hour",
                        "snow_accumulation_06_hour_percentiles",
                        "snow_level",
                        "snow_level_percentiles",
                        "snow_liquid_ratio",
                        "snow_liquid_ratio_percentiles",
                        "spc_hail_04_hour_probability",
                        "spc_tornado_04_hour_probability",
                        "spc_wind_04_hour_probability",
                        "temperature",
                        "temperature_standard_deviation",
                        "thunderstorm_01_hour_probability",
                        "thunderstorm_03_hour_probability",
                        "thunderstorm_06_hour_probability",
                        "thunderstorm_12_hour_probability",
                        "thunderstorm_coverage",
                        "total_cloud_cover",
                        "total_cloud_cover_standard_deviation",
                        "transport_wind_direction",
                        "transport_wind_speed",
                        "ventilation_rate",
                        "vertically_integrated_liquid",
                        "visibility",
                        "visibility_probability",
                        "wind_direction",
                        "wind_gust",
                        "wind_gust_standard_deviation",
                        "wind_speed",
                        "wind_speed_standard_deviation"
                    ],
                    "datetimes":[
                        "2025-09-22T01:00:00",
                        "2025-09-22T02:00:00",
                        "...etc",
                    ],
                    "desi":{
                        "projDict":{
                            "proj":"+proj=lcc +lat_0=25 +lon_0=265 +lat_1=25 +lat_2=
25 +x_0=0 +y_0=0 +R=6371200 +units=m +no_defs +type=crs",
                            "crs_wkt":"PROJCRS[\"unknown\",BASEGEOGCRS[\"unknown\",
DATUM[\"unknown\",ELLIPSOID[\"unknown\",6371200,0,LENGTHUNIT[\"metre\",1,
ID[\"EPSG\",9001]]]],PRIMEM[\"Greenwich\",0,ANGLEUNIT[\"degree\",0.01745
32925199433],ID[\"EPSG\",8901]]],CONVERSION[\"unknown\",METHOD[\"Lambert
Conic Conformal (2SP)\",ID[\"EPSG\",9802]],PARAMETER[\"Latitude of false
origin\",25,ANGLEUNIT[\"degree\",0.0174532925199433],ID[\"EPSG\",8821]],
PARAMETER[\"Longitude of false origin\",265,ANGLEUNIT[\"degree\",0.017
4532925199433],ID[\"EPSG\",8822]],PARAMETER[\"Latitude of 1st standard
parallel\",25,ANGLEUNIT[\"degree\",0.0174532925199433],ID[\"EPSG\",8823]],
PARAMETER[\"Latitude of 2nd standard parallel\",25,ANGLEUNIT[\"degree\",0.0
174532925199433],ID[\"EPSG\",8824]],PARAMETER[\"Easting at false origin\",0,
LENGTHUNIT[\"metre\",1],ID[\"EPSG\",8826]],PARAMETER[\"Northing at false
origin\",0,LENGTHUNIT[\"metre\",1],ID[\"EPSG\",8827]]],CS[Cartesian,2],
AXIS[\"(E)\",east,ORDER[1],LENGTHUNIT[\"metre\",1,ID[\"EPSG\",9001]]],
AXIS[\"(N)\",north,ORDER[2],LENGTHUNIT[\"metre\",1,ID[\"EPSG\",9001]]]]",
                            "first_lat":-263793.73346456443,
                            "first_lon":-3271151.605837171,
                            "nx":2345,
                            "ny":2345,
                            "dx":2539.703,
                            "dy":2539.703
                        }
                    },
                    "parameter-names-dict":{
                        "apparent_temperature":{
                            "attrs":{
                                "grib_section3":[
```

```
                              0,
                              3744965,
                              0,
                              0,
                              30,
                              1,
                              0,
                              6371200,
                              255,
                              255,
                              255,
                              255,
                              2345,
                              1597,
                              19229000,
                              233723400,
                              48,
                              25000000,
                              265000000,
                              2539703,
                              2539703,
                              0,
                              80,
                              25000000,
                              25000000,
                              -90000000,
                              0
                           ],
                           "long_name":"Apparent Temperature",
                           "short_name":"APTMP",
                           "units":"K",
                           "originating_center":"US National Weather Service -
NCEP (WMC)",
                           "originating_sub_center":"NWS Meteorological
Development Laboratory",
                           "master_table_info":"Version Implemented on 7 November
2001",
                           "product_definition_template_number":"Analysis or
forecast at a horizontal level or in a horizontal layer at a point in time.
(see Template 4.0)",
                           "type_of_generating_process":"Forecast",
                           "type_of_first_fixed_surface":"Specified Height Level
Above Ground (m)",
                           "type_of_second_fixed_surface":"Missing (unknown)",
                           "crs_wkt":"PROJCRS[\"unknown\",BASEGEOGCRS[\"unknown\",
DATUM[\"unknown\",ELLIPSOID[\"unknown\",6371200,0,LENGTHUNIT[\"metre\",1,
ID[\"EPSG\",9001]]]],PRIMEM[\"Greenwich\",0,ANGLEUNIT[\"degree\",0.01745
32925199433],ID[\"EPSG\",8901]]],CONVERSION[\"unknown\",METHOD[\"Lambert
Conic Conformal (2SP)\",ID[\"EPSG\",9802]],PARAMETER[\"Latitude of false
origin\",25,ANGLEUNIT[\"degree\",0.0174532925199433],ID[\"EPSG\",8821]],
PARAMETER[\"Longitude of false origin\",265,ANGLEUNIT[\"degree\",0.017
4532925199433],ID[\"EPSG\",8822]],PARAMETER[\"Latitude of 1st standard
parallel\",25,ANGLEUNIT[\"degree\",0.0174532925199433],ID[\"EPSG\",8823]],
PARAMETER[\"Latitude of 2nd standard parallel\",25,ANGLEUNIT[\"degree\",0.0
174532925199433],ID[\"EPSG\",8824]],PARAMETER[\"Easting at false origin\",0,
LENGTHUNIT[\"metre\",1],ID[\"EPSG\",8826]],PARAMETER[\"Northing at false
origin\",0,LENGTHUNIT[\"metre\",1],ID[\"EPSG\",8827]]],CS[Cartesian,2],
AXIS[\"(E)\",east,ORDER[1],LENGTHUNIT[\"metre\",1,ID[\"EPSG\",9001]]],
AXIS[\"(N)\",north,ORDER[2],LENGTHUNIT[\"metre\",1,ID[\"EPSG\",9001]]]]",
                           "gridlength_x_direction":2539.703,
                           "gridlength_y_direction":2539.703,
                           "latitude_first_gridpoint":19.229,
                           "longitude_first_gridpoint":233.7234,
                           "standard_name":"apparent_air_temperature",
                           "coordinates":"forecast_reference_time lead_time
specified_height_level_above_ground",
                           "_FillValue":"AAAAAAAA+H8="
```

REQUIREMENT 11


```
                        },
                        "time":[
                           "2025-09-22T01:00:00",
                           "2025-09-22T02:00:00",
                           "...etc"
                        ]
                     }
                  },
                  "bbox":[
                     -3271151.605837171,
                     -263793.73346456443,
                     2681912.2261628294,
                     3789572.2545354357
                  ]
               }
            }
         ]
      }
```


7.4.5. Location Query

The Location query returns a location defined by a bounding box dependent on the indices of i and j.

Path = {root}/collections/{collectionId}/instances/{instanceId}/locations/{locationId}

Dependencies: OGC API-Environmental Data Retrieval Standard

REQUIREMENT 12

IDENTIFIER	`/req/nwsviz-application/data-query-location`
INCLUDED IN	Requirements class 1: `http://www.example.org/1.0/req/req-class-nwsviz-application`
STATEMENT	All Collections in the Service *SHALL* provide support for **Locations** queries
A	The **Locations** query *SHALL* support the following output formats: • CoverageJSON • Zarr • NetCDF
B	The default output format for the **Locations** query *SHALL* be **CoverageJSON**
C	The **Locations** query *SHALL* support the following HTTP methods: • GET • POST
D	The **Locations** query *SHALL* support a custom query parameter called “ij” where the argument represents index values of the rectangular query.

REQUIREMENT 12	
STATEMENT	`&#x26;ij=i0,j0,i1,j1`

7.4.6. Instances Query

Having multiple versions or instances of the same Collection, where the same information is reprocessed or regenerated is not unusual. Although these versions could be described as new Collections the instance query type allows this data to be described as different views of the same Collection.

Path = {root}/collections/{collectionId}/instances

Dependencies: OGC API-Environmental Data Retrieval Standard

REQUIREMENT 13	
IDENTIFIER	`/req/nwsviz-application/data-query-instances`
INCLUDED IN	Requirements class 1: `http://www.example.org/1.0/req/req-class-nwsviz-application`
STATEMENT	Verify that the Vertical extent demo Collection Instances query is correctly defined
A	Vertical extent demo Collection *SHALL* support Instances of the Collection.
B	Instance ids *SHALL* be an RFC3339 Zulu (UTC)representation of the model run time at a detail level of minutes.
C	Instance ids *SHALL* match the following regular expression ^\d{4}-\d{2}-\d{2}T\d{2}:\d{2}Z$

7.5. General Requirements

7.5.1. HTTP Status Codes

HTTP response

- Response status codes

REQUIREMENT 14

IDENTIFIER	`/req/nwsviz-application/status-codes`
INCLUDED IN	Requirements class 1: `http://www.example.org/1.0/req/req-class-nwsviz-application`
STATEMENT	The NWSViz profile Collection profile *SHALL* support the following HTTP status codes.
A	The message descriptions *SHALL* be as follows:

STATEMENT

STATUS CODE	DESCRIPTION
200	A successful request.
400	The server cannot or will not process the request due to an apparent client error. For example, a query parameter had an incorrect value.
404	The requested resource does not exist on the server. For example, a path parameter had an incorrect value.

REQUIREMENT 15

IDENTIFIER	`/req/nwsviz-application/msg-bodies`
INCLUDED IN	Requirements class 1: `http://www.example.org/1.0/req/req-class-nwsviz-application`
STATEMENT	The NWSViz profile Collection profile *SHALL* support the following HTTP response body.
A	The NWSViz profile Collection *SHALL* use the following JSON schema for error response bodies (HTTP Status Codes 400 and above)

STATEMENT

```
{
    "type": "object",
    "required": [
        "code"
    ],
    "properties": {
        "code": {
            "type": "string"
        },
        "description": {
            "type": "string"
        }
    }
}
```

7.5.2. Links

- Response links

REQUIREMENT 16	
IDENTIFIER	`/req/nwsviz-application/links`
INCLUDED IN	Requirements class 1: `http://www.example.org/1.0/req/req-class-nwsviz-application`
STATEMENT	A NWSViz profile Collection *SHALL* define links as follows:
A	All link objects *SHALL* have **href**, **rel** and **type** attributes
B	The following links *SHALL* be included in the Collection response
STATEMENT	`{ "links":[{ "type":"application/json", "rel":"root", "title":"The landing page of this server as JSON", "href":"https://edr-api-desi-c.mdl.nws.noaa.gov?f=json" }, { "type":"application/json", "rel":"self", "title":"This document as JSON", "href":"https://edr-api-desi-c.mdl.nws.noaa.gov/collections?f=json" }, { "type":"application/json", "rel":"collection", "title":"Collection Metadata for this collection as JSON", "href":"https://edr-api-desi-c.mdl.nws.noaa.gov/collections/NBM_icechunk?f=json" }, { "type":"application/json", "rel":"data", "title":"Items Query for this collection as JSON", "href":"https://edr-api-desi-c.mdl.nws.noaa.gov/collections/NBM_icechunk/items?f=json" }, { "type":"application/json", "rel":"data", "title":"Locations Query for this collection as JSON", "href":"https://edr-api-desi-c.mdl.nws.noaa.gov/collections/NBM_icechunk/locations?f=json" }, { "type":"application/json", "rel":"data", "title":"Position Query for this collection as JSON",`

REQUIREMENT 16


```
            "href":"https://edr-api-desi-c.mdl.nws.noaa.gov/collections/NBM_
icechunk/position?f=json"
         },
         {
            "type":"application/json",
            "rel":"instances",
            "title":"Instances Metadata for this collection as JSON",
            "href":"https://edr-api-desi-c.mdl.nws.noaa.gov/collections/NBM_
icechunk/instances?f=json"
         }
      ]
   }
```

A

ANNEX A (INFORMATIVE) CONFORMANCE CLASS ABSTRACT TEST SUITE (NORMATIVE)

A

ANNEX A (INFORMATIVE) CONFORMANCE CLASS ABSTRACT TEST SUITE (NORMATIVE)

A.1. Conformance Class Core

CONFORMANCE CLASS A.1: CONFORMANCE CLASS 'CORE'	
IDENTIFIER	`http://www.example.org/1.0/conf-class-nwsviz-application`
REQUIREMENTS CLASS	Requirements class 1: `http://www.example.org/1.0/req/req-class-nwsviz-application`
CONFORMANCE TESTS	Abstract test A.2: `/conf/nwsviz-application/root` Abstract test A.1: `/conf/nwsviz-application/edr-conformant` Abstract test A.3: `/conf/nwsviz-application/MRMS-parameter-names` Abstract test A.4: `/conf/nwsviz-application/NBM-parameter-names` Abstract test A.5: `/conf/nwsviz-application/collectionid` Abstract test A.6: `/conf/nwsviz-application/extent` Abstract test A.7: `/conf/nwsviz-application/output-format` Abstract test A.8: `/conf/nwsviz-application/status-codes` Abstract test A.9: `/conf/nwsviz-application/msg-bodies` Abstract test A.10: `/conf/nwsviz-application/links` Abstract test A.11: `/conf/nwsviz-application/data-query` Abstract test A.12: `/conf/nwsviz-application/data-query-instances` Abstract test A.13: `/conf/nwsviz-application/data-query-position` Abstract test A.16: `/conf/nwsviz-application/data-query-items` Abstract test A.14: `/conf/nwsviz-application/data-query-location` Abstract test A.15: `/conf/nwsviz-application/data-query-locations`

ABSTRACT TEST A.1	
IDENTIFIER	`/conf/nwsviz-application/edr-conformant`

ABSTRACT TEST A.1

REQUIREMENT	Requirement 1: `/req/nwsviz-application/edr-conformant`
TEST PURPOSE	Validiate the the NWSViz profile Collection is a valid OGC API-EDR implementation
TEST METHOD	
STEP	Verify that that the Collection can pass the OGC API-EDR Part 1 — Core conformance tests.

ABSTRACT TEST A.2

IDENTIFIER	`/conf/nwsviz-application/root`
REQUIREMENT	Requirement 2: `/req/nwsviz-application_data/root`
TEST PURPOSE	Validate that the profile defines the service landing page
TEST METHOD	
STEP	Verify that the profile defines a **Title** for the service and the value is **"NWSViz profile service"** Verify that each link defined for the service has a **href** and **rel** attribute. Verify that the **Keywords** attribute has the following values: • Geopotential Height • Pressure Level • Specific humidity • Air temperature • u-component of wind • v-component of wind Verify that the **Provider** attribute has **name** and **url** attributes defined. Verify that the **Contact** attribute has an **email** attributes defined. Verify that the **Links** property includes the following link to the OpenAPI definition of the profile:
DESCRIPTION	`{ "title": "OpenAPI definition of NWSViz profile profile", "href": "https://www.example.org/edr/profile/vextdemo/openapi.json", "rel": "profile", "type": "application/json"`

ABSTRACT TEST A.2

```
            }
```

ABSTRACT TEST A.3

IDENTIFIER	`/conf/nwsviz-application/MRMS-parameter-names`
REQUIREMENT	Requirement 6: `/req/nwsviz-application/MRMS-parameter-names`
TEST PURPOSE	Verify that NWSViz profile Collection uses the correct parameter-names schema.
TEST METHOD	
STEP	Verify that the profile Collection follows the following schema where the metadata is dynamically loaded from the underlying store.
DESCRIPTION	

```
{
   "parameter-names":{
      "EchoTop_18":{
         "Type":"Parameter",
         "id":"EchoTop_18",
         "description":"EchoTop_18",
         "temporal":{
            "values":[
               "2025-10-01T18:24:39",
               "2025-10-01T18:32:41",
               "...etc"
            ]
         },
         "attrs":{
            "_FillValue":"AAAAAAAA+H8="
         }
      },
      "EchoTop_30":{
         "Type":"Parameter",
         "id":"EchoTop_30",
         "description":"EchoTop_30",
         "temporal":{
            "values":[
               "2025-10-01T17:58:37",
               "2025-10-01T18:02:40",
               "...etc"
            ]
         },
         "attrs":{
            "_FillValue":"AAAAAAAA+H8="
         }
      },
      "EchoTop_60":{
         "Type":"Parameter",
         "id":"EchoTop_60",
         "description":"EchoTop_60",
         "temporal":{
            "values":[
               "2025-10-01T18:06:38",
               "2025-10-01T18:10:35",
               "...etc"
            ]
         },
         "attrs":{
```

ABSTRACT TEST A.3


```
            "_FillValue":"AAAAAAAA+H8="
         }
      },
      "H50Above0C":{
         "Type":"Parameter",
         "id":"H50Above0C",
         "description":"H50Above0C",
         "temporal":{
            "values":[
               "2025-10-01T18:00:34",
               "2025-10-01T18:04:39",
               "...etc"
            ]
         },
         "attrs":{
            "_FillValue":"AAAAAAAA+H8="
         }
      },
      "H60AboveM20C":{
         "Type":"Parameter",
         "id":"H60AboveM20C",
         "description":"H60AboveM20C",
         "temporal":{
            "values":[
               "2025-10-01T18:26:38",
               "2025-10-01T18:28:40",
               "...etc"
            ]
         },
         "attrs":{
            "_FillValue":"AAAAAAAA+H8="
         }
      },
      "LightningProbabilityNext30minGrid":{
         "Type":"Parameter",
         "id":"LightningProbabilityNext30minGrid",
         "description":"LightningProbabilityNext30minGrid",
         "temporal":{
            "values":[
               "2025-10-01T17:58:37",
               "2025-10-01T18:00:34",
               "2025-10-01T18:02:35",
               "...etc"
            ]
         },
         "attrs":{
            "_FillValue":"AAAAAAAA+H8="
         }
      },
      "POSH":{
         "Type":"Parameter",
         "id":"POSH",
         "description":"POSH",
         "temporal":{
            "values":[
               "2025-10-01T18:12:36",
               "2025-10-01T18:16:39",
               "...etc"
            ]
         },
         "attrs":{
            "_FillValue":"AAAAAAAA+H8="
         }
      },
      "PrecipFlag":{
         "Type":"Parameter",
         "id":"PrecipFlag",
```

ABSTRACT TEST A.3


```
            "description":"PrecipFlag",
            "temporal":{
               "values":[
                  "2025-10-01T18:04:00",
                  "2025-10-01T18:06:00",
                  "...etc"
               ]
            },
            "attrs":{
               "_FillValue":"AAAAAAAA+H8="
            }
         },
         "RadarOnly_QPE_01H":{
            "Type":"Parameter",
            "id":"RadarOnly_QPE_01H",
            "description":"RadarOnly_QPE_01H",
            "temporal":{
               "values":[
                  "2025-10-01T17:58:00",
                  "2025-10-01T18:00:00",
                  "...etc"
               ]
            },
            "attrs":{
               "_FillValue":"AAAAAAAA+H8="
            }
         },
         "RadarOnly_QPE_03H":{
            "Type":"Parameter",
            "id":"RadarOnly_QPE_03H",
            "description":"RadarOnly_QPE_03H",
            "temporal":{
               "values":[
                  "2025-10-01T18:00:00"
               ]
            },
            "attrs":{
               "_FillValue":"AAAAAAAA+H8="
            }
         },
         "RadarOnly_QPE_15M":{
            "Type":"Parameter",
            "id":"RadarOnly_QPE_15M",
            "description":"RadarOnly_QPE_15M",
            "temporal":{
               "values":[
                  "2025-10-01T18:00:00",
                  "2025-10-01T18:15:00",
                  "...etc"
               ]
            },
            "attrs":{
               "_FillValue":"AAAAAAAA+H8="
            }
         },
         "RadarOnly_QPE_72H":{
            "Type":"Parameter",
            "id":"RadarOnly_QPE_72H",
            "description":"RadarOnly_QPE_72H",
            "temporal":{
               "values":[
                  "2025-10-01T18:00:00"
               ]
            },
            "attrs":{
               "_FillValue":"AAAAAAAA+H8="
            }
```

ABSTRACT TEST A.3


```
                    },
                    "Reflectivity_-10C":{
                       "Type":"Parameter",
                       "id":"Reflectivity_-10C",
                       "description":"Reflectivity_-10C",
                       "temporal":{
                          "values":[
                             "2025-10-01T18:02:35",
                             "2025-10-01T18:06:38",
                             "...etc"
                          ]
                       },
                       "attrs":{
                          "_FillValue":"AAAAAAAA+H8="
                       }
                    },
                    "Reflectivity_-20C":{
                       "Type":"Parameter",
                       "id":"Reflectivity_-20C",
                       "description":"Reflectivity_-20C",
                       "temporal":{
                          "values":[
                             "2025-10-01T18:20:37",
                             "2025-10-01T18:22:35",
                             "...etc"
                          ]
                       },
                       "attrs":{
                          "_FillValue":"AAAAAAAA+H8="
                       }
                    },
                    "Reflectivity_0C":{
                       "Type":"Parameter",
                       "id":"Reflectivity_0C",
                       "description":"Reflectivity_0C",
                       "temporal":{
                          "values":[
                             "2025-10-01T17:58:37",
                             "2025-10-01T18:00:34",
                             "...etc"
                          ]
                       },
                       "attrs":{
                          "_FillValue":"AAAAAAAA+H8="
                       }
                    },
                    "VII":{
                       "Type":"Parameter",
                       "id":"VII",
                       "description":"VII",
                       "temporal":{
                          "values":[
                             "2025-10-01T18:00:34",
                             "2025-10-01T18:02:40",
                             "...etc"
                          ]
                       },
                       "attrs":{
                          "_FillValue":"AAAAAAAA+H8="
                       }
                    },
                    "VIL":{
                       "Type":"Parameter",
                       "id":"VIL",
                       "description":"VIL",
                       "temporal":{
                          "values":[
```

ABSTRACT TEST A.3

```
                        "2025-10-01T18:00:34",
                        "2025-10-01T18:04:39",
                        "...etc"
                     ]
                  },
                  "attrs":{
                     "_FillValue":"AAAAAAAA+H8="
                  }
               }
            }
         }
```

ABSTRACT TEST A.4

IDENTIFIER	`/conf/nwsviz-application/NBM-parameter-names`
REQUIREMENT	Requirement 5: `/req/nwsviz-application/NBM-parameter-names`
TEST PURPOSE	Verify that NWSViz profile Collection uses the correct parameter-names schema.
TEST METHOD	
STEP	Verify that the profile Collection follows the following schema where the metadata is dynamically loaded from the underlying store.
DESCRIPTION	

```
"parameter-names":{
   "apparent_temperature":{
      "Type":"Parameter",
      "id":"apparent_temperature",
      "description":"apparent_temperature",
      "unit":{
         "symbol":"K"
      },
      "temporal":{
         "values":[
            "2025-09-22T01:00:00",
            "2025-09-22T02:00:00",
            "...etc"
         ]
      },
      "attrs":{
         "grib_section3":[
            0,
            3744965,
            0,
            0,
            30,
            1,
            0,
            6371200,
            255,
            255,
            255,
            255,
            2345,
            1597,
            19229000,
            233723400,
            48,
```

ABSTRACT TEST A.4


```
                  25000000,
                  265000000,
                  2539703,
                  2539703,
                  0,
                  80,
                  25000000,
                  25000000,
                  -90000000,
                  0
               ],
               "long_name":"Apparent Temperature",
               "short_name":"APTMP",
               "units":"K",
               "originating_center":"US National Weather Service - NCEP (WMC)",
               "originating_sub_center":"NWS Meteorological Development
            Laboratory",
               "master_table_info":"Version Implemented on 7 November 2001",
               "product_definition_template_number":"Analysis or forecast at
            a horizontal level or in a horizontal layer at a point in time. (see
            Template 4.0)",
               "type_of_generating_process":"Forecast",
               "type_of_first_fixed_surface":"Specified Height Level Above
            Ground (m)",
               "type_of_second_fixed_surface":"Missing (unknown)",
               "crs_wkt":"PROJCRS[\"unknown\",BASEGEOGCRS[\"unknown\",
            DATUM[\"unknown\",ELLIPSOID[\"unknown\",6371200,0,LENGTHUNIT[\"metre\",1,
            ID[\"EPSG\",9001]]]],PRIMEM[\"Greenwich\",0,ANGLEUNIT[\"degree\",0.01745
            32925199433],ID[\"EPSG\",8901]]],CONVERSION[\"unknown\",METHOD[\"Lambert
            Conic Conformal (2SP)\",ID[\"EPSG\",9802]],PARAMETER[\"Latitude of
            false origin\",25,ANGLEUNIT[\"degree\",0.0174532925199433],ID[\"EPSG
            \",8821]],PARAMETER[\"Longitude of false origin\",265,ANGLEUNIT[\"d
            egree\",0.0174532925199433],ID[\"EPSG\",8822]],PARAMETER[\"Latitude
            of 1st standard parallel\",25,ANGLEUNIT[\"degree\",0.01745329251994
            33],ID[\"EPSG\",8823]],PARAMETER[\"Latitude of 2nd standard parallel
            \",25,ANGLEUNIT[\"degree\",0.0174532925199433],ID[\"EPSG\",8824]],
            PARAMETER[\"Easting at false origin\",0,LENGTHUNIT[\"metre\",1],ID[\"EPSG
            \",8826]],PARAMETER[\"Northing at false origin\",0,LENGTHUNIT[\"metre
            \",1],ID[\"EPSG\",8827]]],CS[Cartesian,2],AXIS[\"(E)\",east,ORDER[1],
            LENGTHUNIT[\"metre\",1,ID[\"EPSG\",9001]]],AXIS[\"(N)\",north,ORDER[2],
            LENGTHUNIT[\"metre\",1,ID[\"EPSG\",9001]]]]",
               "gridlength_x_direction":2539.703,
               "gridlength_y_direction":2539.703,
               "latitude_first_gridpoint":19.229,
               "longitude_first_gridpoint":233.7234,
               "standard_name":"apparent_air_temperature",
               "coordinates":"forecast_reference_time lead_time specified_
            height_level_above_ground",
               "_FillValue":"AAAAAAAA+H8="
            }
         }
      }
```


ABSTRACT TEST A.5

IDENTIFIER	/conf/nwsviz-application/collectionid
REQUIREMENT	Requirement 3: /req/nwsviz-application/collectionid

ABSTRACT TEST A.5

TEST PURPOSE	Validate that a collectionid requirement is correctly defined.
TEST METHOD	
STEP	Verify that collectionids match the following:

DESCRIPTION

id	description
NBM_icechunk	National Blend of Models
MRMS_icechunk	Multi-Radar/Multi-Sensor Radar Products

ABSTRACT TEST A.6

IDENTIFIER	`/conf/nwsviz-application/extent`
REQUIREMENT	Requirement 4: `/req/nwsviz-application/extent`
TEST PURPOSE	Verify that NWSViz profile Collection Extent is correctly defined in the Collection response.
TEST METHOD	
STEP	Verify that the collection contains a spatial extent and contain an additional attribute named “locations” containing the a list of the available locations within the collection.

DESCRIPTION

```
{
   "spatial":{
      "bbox":[
         -180,
         -90,
         180,
         90
      ],
      "locations":[
         "CO",
         "AK",
         "HI"
      ]
   }
}
```

part	Verify that the Profile contains a temporal extent containing an interval and values.
part	Verify that the Profile contains an attribute named instances which contains a list of available instances for the collection.

ABSTRACT TEST A.6

	part	Verify that the Profile contains a custom query dimension with the name “ensemble” containing values and ids.
	part	Verify that the Profile contains a vertical extent if applicable

ABSTRACT TEST A.7

IDENTIFIER	`/conf/nwsviz-application/output-format`
REQUIREMENT	Requirement 8: `/req/nwsviz-application/output-format`
TEST PURPOSE	Verify that NWSViz profile Collection queries format data responses correctly.
TEST METHOD	
STEP	Request data with the output format label and verify the response is correct
	Verify a output format of **GRIB2** creates a file that has the following structure https://library.wmo.int/viewer/35625?medianame=306_v.I.2_2019_edition_Updated_2022_en_#page=31&viewer=picture
A	Verify a output format of **GeoJSON** is compatible with the JSON schema defined in the following repository https://github.com/opengeospatial/OGC-feat-geo-json
B	Verify a output format of **CoverageJSON** is compatible with the JSON schema defined by https://docs.ogc.org/cs/21-069r2/21-069r2.html
C	Verify a output format of **NetCDF4** creates a file that has the structure described by https://cfconventions.org/Data/cf-conventions/cf-conventions-1.10/cf-conventions.html
D	Verify a output format of **CSV** generates a CSV output file with a metadata file which is described by https://www.w3.org/TR/2015/REC-tabular-data-model-20151217/

ABSTRACT TEST A.8

IDENTIFIER	`/conf/nwsviz-application/status-codes`
REQUIREMENT	Requirement 14: `/req/nwsviz-application/status-codes`
TEST PURPOSE	Verify that NWSViz profile Collection has the correct HTTP status responses.
TEST METHOD	

ABSTRACT TEST A.8

STEP	Verify that a valid request returns a HTTP status code of 200 Verify that a valid request with an empty response returns a HTTP status code of 204 Verify that a request with an invalid query parameter value returns a HTTP status code of 400 Verify that that an unsupported HTTP method request returns a HTTP status code of 405. Verify that a request for too much data returns a HTTP status code of 413. Verify that a invalid request caused by an unexpected error returns a HTTP status code of 500.

ABSTRACT TEST A.9

IDENTIFIER	`/conf/nwsviz-application/msg-bodies`
REQUIREMENT	Requirement 15: `/req/nwsviz-application/msg-bodies`
TEST PURPOSE	Verify that NWSViz profile error response messages have the correct structure.
TEST METHOD	
STEP	Verify the the message body returned by error messages (status codes 400,405,413 and 500) is JSON based on the following schema:
DESCRIPTION	`{ "type": "object", "required": [ "code" ], "properties": { "code": { "type": "string" }, "description": { "type": "string" } } }`

ABSTRACT TEST A.10

IDENTIFIER	`/conf/nwsviz-application/links`
REQUIREMENT	Requirement 16: `/req/nwsviz-application/links`

ABSTRACT TEST A.10

TEST PURPOSE	Verify that NWSViz profile Collection Links section contains the required links
TEST METHOD	
STEP	Verify that all links in the NWSViz profile Collection have **href**, **rel** and **type** attributes Verify that NWSViz profile Collection Links section contains the required links:
DESCRIPTION	`{ "href": "https://creativecommons.org/licenses/by-nc/4.0/", "rel": "licence", "type": "text/html" }, { "href": "https://library.wmo.int/viewer/35625?medianame=306_v.I.2_2019_edition_Updated_2022_en_#page=31&viewer=picture", "rel": "related", "title": "FM 92-XIV GRIB specification", "type": "text/html" }`

ABSTRACT TEST A.11

IDENTIFIER	`/conf/nwsviz-application/data-query`
REQUIREMENT	Requirement 7: `/req/nwsviz-application/data-query`
TEST PURPOSE	Verify that all required NWSViz profile Collection queries are defined in the Collection response.
TEST METHOD	
STEP	Verify that the Collection response includes definitions for the following queries. • Position • Cube • Locations • Instances

ABSTRACT TEST A.12

IDENTIFIER	`/conf/nwsviz-application/data-query-instances`
REQUIREMENT	Requirement 13: `/req/nwsviz-application/data-query-instances`
TEST PURPOSE	Verify that NWSViz profile Collection **Instances** query is correctly defined in the Collection response.

ABSTRACT TEST A.12

TEST METHOD	
STEP	Verify that NWSViz profile Collection has a **Instances** data query defined. Verify the **Instances** ids str against the Regular expression ^\d{4}-\d{2}-\d{2}T\d{2}:\d{2}Z$

ABSTRACT TEST A.13

IDENTIFIER	`/conf/nwsviz-application/data-query-position`
REQUIREMENT	Requirement 9: `/req/nwsviz-application/data-query-position`
TEST PURPOSE	Verify that NWSViz profile Collection **Position** query is correctly defined in the Collection response.
TEST METHOD	
STEP	Verify that NWSViz profile Collection **Position** query defines the following output_format types • CoverageJSON Verify that NWSViz profile Collection **Position** query defines the following default output_format. • CoverageJSON Verify that NWSViz profile Collection **Position** requirement supports the following HTTP operations • GET • POST

ABSTRACT TEST A.14

IDENTIFIER	`/conf/nwsviz-application/data-query-location`
REQUIREMENT	Requirement 12: `/req/nwsviz-application/data-query-location`
TEST PURPOSE	Verify that NWSViz profile Collection **Location** query is correctly defined in the Collection response.
TEST METHOD	
STEP	Verify that NWSViz profile Collection **Location** query defines the following output_format types

ABSTRACT TEST A.14

	• CoverageJSON • Zarr • NetCDF Verify that NWSViz profile Collection **Location** query defines the following default output_format. • CoverageJSON Verify that NWSViz profile Collection **Location** requirement supports the following HTTP operations • GET • POST Verify that NWSViz profile Collection **Location** requirement supports the following custom query dimension
DESCRIPTION	`&#x26;ij=i0,j0,i1,j1`

ABSTRACT TEST A.15

IDENTIFIER	`/conf/nwsviz-application/data-query-locations`
REQUIREMENT	Requirement 11: `/req/nwsviz-application/data-query-locations`
TEST PURPOSE	Verify that NWSViz profile Collection **Locations** query is correctly defined in the Collection response.
TEST METHOD	
STEP	Verify that NWSViz profile Collection **Locations** query defines the following output_format types • GeoJSON Verify that NWSViz profile Collection **Locations** query defines the following default output_format. • CoverageJSON Verify that NWSViz profile Collection **Locations** requirement supports the following HTTP operations

ABSTRACT TEST A.15

- GET
- POST

Verify that the reponse provides a listing of the available locations in a collection that conforms to GeoJSON and contains the properties contained in this example:

DESCRIPTION


```
{
   "type":"FeatureCollection",
   "features":[
      {
         "type":"Feature",
         "geometry":{
            "type":"Polygon",
            "coordinates":[
               [
                  [
                     2681912.2261628294,
                     -263793.73346456443
                  ],
                  [
                     2681912.2261628294,
                     3789572.2545354357
                  ],
                  [
                     -3271151.605837171,
                     3789572.2545354357
                  ],
                  [
                     -3271151.605837171,
                     -263793.73346456443
                  ],
                  [
                     2681912.2261628294,
                     -263793.73346456443
                  ]
               ]
            ]
         },
         "properties":{
            "instance":"2025-09-22T00:00:00",
            "name":"conus",
            "name_alt":"link:++http://edr-api-desi-c.mdl.nws.noaa.gov/
collections/NBM_icechunk/instances/2025-09-22T00:00:00/locations/conus?f=
json++[]",
            "edrqueryendpoint":"link:++http://edr-api-desi-c.mdl.nws.noaa.
gov/collections/NBM_icechunk/instances/2025-09-22T00:00:00/locations/
conus?f=json++[]",
            "parameter-names":[
               "apparent_temperature",
               "ceiling",
               "ceiling_probability",
               "cloud_base",
               "conditional_probability_of_precipitation_type",
               "convective_available_potential_energy",
               "convective_available_potential_energy_percentiles",
               "convective_available_potential_energy_standard_deviation",
               "dewpoint",
               "dewpoint_standard_deviation",
               "downward_shortwave_radiation_flux",
               "dry_thunderstorm_probability",
               "echo_top",
               "ellrod_index",
               "fosberg_index_06_hour",
```

ABSTRACT TEST A.15


```
                        "ice_accumulation_01_hour",
                        "ice_accumulation_06_hour",
                        "ice_accumulation_06_hour_percentiles",
                        "low_level_turbulence",
                        "low_level_wind_shear_altitude",
                        "low_level_wind_shear_direction",
                        "low_level_wind_shear_magnitude",
                        "maximum_reflectivity",
                        "maximum_relative_humidity_12_hour",
                        "maximum_temperature_12_hour",
                        "maximum_temperature_12_hour_standard_deviation",
                        "minimum_relative_humidity_12_hour",
                        "minimum_temperature_12_hour",
                        "minimum_temperature_12_hour_standard_deviation",
                        "mixing_height",
                        "precipitable_water",
                        "precipitable_water_percentiles",
                        "precipitation_01_hour_probability",
                        "precipitation_06_hour_probability",
                        "precipitation_12_hour_probability",
                        "precipitation_accumulation_01_hour",
                        "precipitation_accumulation_06_hour",
                        "precipitation_duration_12_hour",
                        "predominant_weather",
                        "relative_humidity",
                        "sea_surface_temperature",
                        "snow_accumulation_01_hour",
                        "snow_accumulation_01_hour_percentiles",
                        "snow_accumulation_06_hour",
                        "snow_accumulation_06_hour_percentiles",
                        "snow_level",
                        "snow_level_percentiles",
                        "snow_liquid_ratio",
                        "snow_liquid_ratio_percentiles",
                        "spc_hail_04_hour_probability",
                        "spc_tornado_04_hour_probability",
                        "spc_wind_04_hour_probability",
                        "temperature",
                        "temperature_standard_deviation",
                        "thunderstorm_01_hour_probability",
                        "thunderstorm_03_hour_probability",
                        "thunderstorm_06_hour_probability",
                        "thunderstorm_12_hour_probability",
                        "thunderstorm_coverage",
                        "total_cloud_cover",
                        "total_cloud_cover_standard_deviation",
                        "transport_wind_direction",
                        "transport_wind_speed",
                        "ventilation_rate",
                        "vertically_integrated_liquid",
                        "visibility",
                        "visibility_probability",
                        "wind_direction",
                        "wind_gust",
                        "wind_gust_standard_deviation",
                        "wind_speed",
                        "wind_speed_standard_deviation"
                    ],
                    "datetimes":[
                        "2025-09-22T01:00:00",
                        "2025-09-22T02:00:00",
                        "...etc",
                    ],
                    "desi":{
                        "projDict":{
                            "proj":"+proj=lcc +lat_0=25 +lon_0=265 +lat_1=25 +lat_2=
            25 +x_0=0 +y_0=0 +R=6371200 +units=m +no_defs +type=crs",
```

```
                    "crs_wkt":"PROJCRS[\"unknown\",BASEGEOGCRS[\"unknown\",
DATUM[\"unknown\",ELLIPSOID[\"unknown\",6371200,0,LENGTHUNIT[\"metre\",1,
ID[\"EPSG\",9001]]]],PRIMEM[\"Greenwich\",0,ANGLEUNIT[\"degree\",0.01745
32925199433],ID[\"EPSG\",8901]]],CONVERSION[\"unknown\",METHOD[\"Lambert
Conic Conformal (2SP)\",ID[\"EPSG\",9802]],PARAMETER[\"Latitude of
false origin\",25,ANGLEUNIT[\"degree\",0.0174532925199433],ID[\"EPSG
\",8821]],PARAMETER[\"Longitude of false origin\",265,ANGLEUNIT[\"d
egree\",0.0174532925199433],ID[\"EPSG\",8822]],PARAMETER[\"Latitude
of 1st standard parallel\",25,ANGLEUNIT[\"degree\",0.01745329251994
33],ID[\"EPSG\",8823]],PARAMETER[\"Latitude of 2nd standard parallel
\",25,ANGLEUNIT[\"degree\",0.0174532925199433],ID[\"EPSG\",8824]],
PARAMETER[\"Easting at false origin\",0,LENGTHUNIT[\"metre\",1],ID[\"EPSG
\",8826]],PARAMETER[\"Northing at false origin\",0,LENGTHUNIT[\"metre
\",1],ID[\"EPSG\",8827]]],CS[Cartesian,2],AXIS[\"(E)\",east,ORDER[1],
LENGTHUNIT[\"metre\",1,ID[\"EPSG\",9001]]],AXIS[\"(N)\",north,ORDER[2],
LENGTHUNIT[\"metre\",1,ID[\"EPSG\",9001]]]]",
                    "first_lat":-263793.73346456443,
                    "first_lon":-3271151.605837171,
                    "nx":2345,
                    "ny":2345,
                    "dx":2539.703,
                    "dy":2539.703
                 }
              },
              "parameter-names-dict":{
                 "apparent_temperature":{
                    "attrs":{
                       "grib_section3":[
                          0,
                          3744965,
                          0,
                          0,
                          30,
                          1,
                          0,
                          6371200,
                          255,
                          255,
                          255,
                          255,
                          2345,
                          1597,
                          19229000,
                          233723400,
                          48,
                          25000000,
                          265000000,
                          2539703,
                          2539703,
                          0,
                          80,
                          25000000,
                          25000000,
                          -90000000,
                          0
                       ],
                       "long_name":"Apparent Temperature",
                       "short_name":"APTMP",
                       "units":"K",
                       "originating_center":"US National Weather Service -
NCEP (WMC)",
                       "originating_sub_center":"NWS Meteorological
Development Laboratory",
                       "master_table_info":"Version Implemented on 7
November 2001",
```

ABSTRACT TEST A.15


```
                        "product_definition_template_number":"Analysis or
forecast at a horizontal level or in a horizontal layer at a point in
time. (see Template 4.0)",
                        "type_of_generating_process":"Forecast",
                        "type_of_first_fixed_surface":"Specified Height
Level Above Ground (m)",
                        "type_of_second_fixed_surface":"Missing (unknown)",
                        "crs_wkt":"PROJCRS[\"unknown\",BASEGEOGCRS[\"unknown
\",DATUM[\"unknown\",ELLIPSOID[\"unknown\",6371200,0,LENGTHUNIT[\"metre
\",1,ID[\"EPSG\",9001]]]],PRIMEM[\"Greenwich\",0,ANGLEUNIT[\"degr
ee\",0.0174532925199433],ID[\"EPSG\",8901]]],CONVERSION[\"unknown
\",METHOD[\"Lambert Conic Conformal (2SP)\",ID[\"EPSG\",9802]],
PARAMETER[\"Latitude of false origin\",25,ANGLEUNIT[\"degree\",0.0174
532925199433],ID[\"EPSG\",8821]],PARAMETER[\"Longitude of false origin
\",265,ANGLEUNIT[\"degree\",0.0174532925199433],ID[\"EPSG\",8822]],
PARAMETER[\"Latitude of 1st standard parallel\",25,ANGLEUNIT[\"degree
\",0.0174532925199433],ID[\"EPSG\",8823]],PARAMETER[\"Latitude of 2nd
standard parallel\",25,ANGLEUNIT[\"degree\",0.0174532925199433],ID[\"EPSG
\",8824]],PARAMETER[\"Easting at false origin\",0,LENGTHUNIT[\"metre
\",1],ID[\"EPSG\",8826]],PARAMETER[\"Northing at false origin\",0,
LENGTHUNIT[\"metre\",1],ID[\"EPSG\",8827]]],CS[Cartesian,2],AXIS[\"(E)\",
east,ORDER[1],LENGTHUNIT[\"metre\",1,ID[\"EPSG\",9001]]],AXIS[\"(N)\",
north,ORDER[2],LENGTHUNIT[\"metre\",1,ID[\"EPSG\",9001]]]]",
                        "gridlength_x_direction":2539.703,
                        "gridlength_y_direction":2539.703,
                        "latitude_first_gridpoint":19.229,
                        "longitude_first_gridpoint":233.7234,
                        "standard_name":"apparent_air_temperature",
                        "coordinates":"forecast_reference_time lead_time
specified_height_level_above_ground",
                        "_FillValue":"AAAAAAAA+H8="
                     },
                     "time":[
                        "2025-09-22T01:00:00",
                        "2025-09-22T02:00:00",
                        "...etc"
                     ]
                  }
               },
               "bbox":[
                  -3271151.605837171,
                  -263793.73346456443,
                  2681912.2261628294,
                  3789572.2545354357
               ]
            }
         }
      ]
}
```


ABSTRACT TEST A.16

IDENTIFIER	`/conf/nwsviz-application/data-query-items`
REQUIREMENT	Requirement 10: `/req/nwsviz-application/data-query-items`
TEST PURPOSE	Verify that NWSViz profile Collection **Items** query is correctly defined in the Collection response.

ABSTRACT TEST A.16

TEST METHOD

STEP Verify that the **Items** query returns a connection to a Virtual Zarr Store that can be accessed via Xarray, such as:

```
import xarray as xr

ds=xr.open_zarr('link:++https://edr-api-desi-c.mdl.nws.noaa.gov/
collections/MRMS_icechunk/instances/2025-10-01T18:00:00/items/zarr/
EchoTop_30/0++[]')
# Where EchoTop_30 is the element name and 0 is the zoom level
```

DESCRIPTION

step	Verify that the **Items** query supports the following HTTP methods: • GET
step	Verify that the **Items** query has an item identifier that follows the following syntax: • zarr/<element>/<zoom>/<crs>/<unit>

B

ANNEX B (INFORMATIVE) REVISION HISTORY

ANNEX B (INFORMATIVE) REVISION HISTORY

Table B.1

DATE	RELEASE	EDITOR	PRIMARY CLAUSES MODIFIED	DESCRIPTION
2016-04-28	0.1	G. Editor	all	initial version

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] *OGC: OGC Testbed 12 Annex B: Architecture* (2015).