Open Geospatial Consortium

Publication Date: 2013-06-18

Approval Date: 2013-04-23

External identifier of this OGC® document: http://www.opengis.net/doc/bp/Name-type-crs

Reference number of this Document: OGC 11-135r2

Version: 1.0

Category: OGC® Best Practice

Editor: Peter Baumann

OGC® Name Type Specification for Coordinate Reference Systems

This document defines an OGC Best Practices 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. This document is subject to change without notice. This document is an official position of the OGC membership on this particular technology topic.

Document type:	OGC Best Practice
Document subtype:	Name Type Definition
Document stage:	Final
Document language:	English

Contents

Page

Tables

Page

i. Abstract

This document specifies a Name Type Specification (NTS) for predefined, combined, and parameterized Coordinate Reference System (CRS) definitions. This NTS augments the `/def/` namespace with http URI definitions for CRSs. The NTS is based on the Name Type Specification – definitions – part 1 – basic name [OGC 09-048r3] and supersedes OGC document "Definition identifier URNs in OGC namespace" [OGC 07-092r3].

NTSs are maintained by the OGC Naming Authority (OGC-NA).

This document includes one Annex: a user guide to the OGC CRS resolver.

ii. Keywords

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

Coordinate Reference System, CRS, CRS identifiers, EPSG

iii. 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.

v. Submitting Organizations

The following organizations submitted this Document to the Open Geospatial Consortium Inc.

□ Jacobs University Bremen	□ rasdaman GmbH

vi. Submitters

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

Name	Organization
Peter Baumann	Jacobs University Bremen, rasdaman GmbH
Dimitar Misev	Jacobs University Bremen, rasdaman GmbH
Jinsongdi Yu	Jacobs University Bremen, rasdaman GmbH
Piero Campalani	Jacobs University Bremen

vii. Future Work

Among the topics for future development are the following items:

- Eventually move this document to Policy Standard status
- Extend CRS URLs with CRS concatenation capabilities
- Investigate on the need of URL identifiers for Coordinate System, datum, ellipsoid, prime meridian, etc.
- Change Request CR to GML to establish EnvelopeWithCRSType (similar to EnvelopeWithTimePeriodType).

OGC® Name Type Specification for Coordinate Reference Systems

1 Scope

This OGC Best Practice document defines a syntax, based on http URIs, for identifying Coordinate Reference Systems (CRSs), including compound and parameterized CRSs. Strings conforming to this syntax are called *CRS URIs*. The semantics of such definitions is established by the OGC registry service. The specification on hand does not define concrete URIs, it only establishes an URI scheme; definition of CRS URIs based on this scheme, such as temporal CRSs, is done separately within OGC.

As OGC's mission centers around location-based data and services, practically all OGC specifications make heavy use of CRSs to relate object locations in space and time to some reference position. A vast amount of different definitions is in practical use, covering horizontal (geodetic) extents, height, and time. New definitions arise, such as in planetary sciences.

For all of these CRSs, a simple, expressive, and http compatible mechanism is required allowing software to create, identify, and understand CRSs and their definitions. This document establishes a convention for naming CRSs within OGC. In particular, the scheme supports the following use cases:

- CRSs with a normatively fixed definition are addressable by a unique URI; e.g., the WGS84 CRS can be identified through http://www.opengis.net/def/crs/EPSG/0/4326;
- Shorthands are possible for enumerating a family of supported CRSs; for example, a server supporting the complete EPSG list of CRSs should be able to express this with one URI, such as http://www.opengis.net/def/crs/EPSG/0/, rather than having to enumerate the whole list of thousands of CRSs;
- Ad-hoc derivations from CRSs are expressible (see, e.g., the WCS Core), such as the result of an x/t slicing through an x/y/t datacube through a WCS or WCPS operation;
- Spatio-temporal CRSs can be established by combining existing CRS definitions; for example, 3-D georeferenced image timeseries can be described through a combination of 2-D WGS84 CRS and some 1-D time CRS.
- Coordinate system axes can be referenced using their names as defined in the CRSs on hand, such as *latitude*, in operations like spatio-temporal subsetting, thereby enabling semantic checks for admissible axis names by services.
- The naming scheme is, without reservation, backwards compatible to OGC's pre-existing URL scheme.

- The naming scheme does not prejudicate any particular CRS definitions, but remains open to any kind of CRSs that may appear in future.
- The naming scheme does not assume one central CRS resolver but supports any number of independently maintained CRS resolvers.

This CRS URI naming scheme is implemented by the OGC CRS resolver[1] since March 2013.

2 Conformance

For the sake of clarity, this document follows the OGC modular specification style; requirements highlighted in grey boxes constitute the normative information. However, as this document is not an OGC standard it does not contain a corresponding Abstract Test Suite (ATS) Annex.

This document establishes one requirements class, *crs-nts*, of URI http://www.opengis.net/spec/CRS-NTS/1.0/req/crs-nts. The complete specification consists of the document, accessible under this URL, plus the XML Schema available from http://schemas.opengis.net/schemas/crs-nts/.

Standardisation target of all requirements and conformance classes are HTTP URIs.

Requirements defined in this document are relative to http://www.opengis.net/spec/CRS-NTS/1.0/.

3 Normative references

An OGC name may be provided for a *definition* of a type of object broadly classified as a "concept" or system parameter. The precise scope of definitions that may be identified with OGC Names is provided by the set of items in the register at http://urn.opengis.net/register/OGC-NA/deftype.

This Name Type Specification for CRSs is available for download from http://www.opengeospatial.org/standards/bp/.

The following normative documents contain provisions that, through reference in this text, constitute provisions of this specification. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. For undated references, the latest edition of the normative document referred to applies.

3.1 Normative references to OGC documents

OGC 08-015r2, *The OpenGIS Abstract Specification, Topic 2: Spatial Referencing by Coordinates,* version 4.0.

OGC 09-046r2, OGC Naming Authority - Procedures

OGC 09-048r3, *Name type specification – definitions – part 1 – basic name*, version 1.1

[1] The source code of this resolver, SECORE (Semantic Resolver for Coordinate Reference Systems), is available as open source from www.rasdaman.org.

OGC 07-036, Geography Markup Language (GML) Encoding Standard, version 3.2.1

OGC 09-110r4, *OGC Web Coverage Service 2.0 Interface Standard – Core*, version 2.0.

OGC 11-053, OGC *Web Coverage Service 2.0 Interface Standard – CRS Extension*, version 1.0.

3.2 Normative references external to OGC

[1] ISO 8601:2004(E) *Data elements and interchange formats — Information interchange — Representation of dates and time*

[2] www.epsg.org

[3] IETF RFC 2396 (August 1998), *Uniform Resource Identifiers (URI): Generic Syntax*, http://www.ietf.org/rfc/rfc2396.txt

[4] IETF RFC 5234 *Augmented BNF for Syntax Specifications: ABNF* (2008). http://tools.ietf.org/html/rfc5234

[5] JSR-233 *Scripting for the Java Platform*. http://jcp.org/aboutJava/communityprocess/final/jsr223/

[6] W3C, *XQuery 1.0: An XML Query Language (Second Edition)*, W3C Recommendation, 14 December 2010. http://www.w3.org/TR/xquery/

4 Name assignment policy

4.1 Document types

The register of document types http://www.opengis.net/register/ogc-na/doc-type is controlled by OGC-NA. Changes to this register (additions, deletions, and supercession) shall be initiated by a submission to the OGC Naming Authority names@opengeospatial.org.

4.2 Document element types

The registers of document element types is controlled by OGC-NA. Changes to this register (additions, deletions, and supercession) shall be by application to ogcna@lists.opengeospatial.org. The current state of the register of document element types is shown at http://www.opengis.net/register/ogc-na/doc-element-type.

4.3 Names

The register of names http://www.opengis.net/register/ogc-na/name is controlled by OGC-NA. Changes to this register (additions, deletions, and supercession) shall be initiated by a submission to the OGC Naming Authority names@opengeospatial.org .

Note The approval of any new public document by OGC shall automatically trigger a registration request to OGC-NA. The name assigned shall be included on the cover page of the document.

5 Terms and definitions

For the purposes of this document, the terms and definitions given in Clause 3 apply and, in addition, the following terms and definitions. An arrow "□" indicates that the term following it is defined elsewhere in this Clause 5.

5.1 Axis definition

GML fragment whose root is in the substitution group of `gml:CoordinateSystemAxis` as defined in [OGC 07-036]

5.2 Axis identifier

URI resolving to exactly one well-known □*Axis definition*

5.3 CRS

Coordinate Reference System

5.4 CRS axis

fixed reference line of a □*CRS* identified by its □Axis identifier and described in its □ Axis definition

5.5 CRS definition

GML fragment whose root is in the substitution group of `gml:AbstractCRS` as defined in [OGC 07-036]

5.6 CRS Identifier

URI resolving to exactly one well-known □*CRS definition*

Note This can be either a CRS in the substitution group of type `gml:AbstractCRS` or a Parameterized CRS in the substitution group of `crsnts:ParameterizedCRS` where all parameters are saturated either via values provided in the URI or through default values provided in the Parameterized CRS definition. See Clause 10.

Example The WGS84 CRS has this CRS Identifier registered with OGC:
http://www.opengis.net/def/crs/EPSG/0/4326
This is equivalent to:
http://www.opengis.net/def/crs?authority=EPSG&version=0&code4326
See Clause 7 for the syntax and semantics definition.

5.7 Compound CRS

sequence of two or more single □*CRS*s

Note Several constraints need to be fulfilled by a compound CRS (which is taken from OGC Abstract Topic 2 [OGC 08-015r2]); however, as the detailed CRS semantics is not a focus of this specification, these constraints are not addressed here.

5.8 Parameterized CRS

□*CRS* boilerplate which, when provided with concrete values for all its formal parameters, resembles a □*CRS definition*

6 Conventions

6.1 Namespace prefix conventions

The following namespaces are used in this document. The prefix abbreviations used constitute conventions used here, but are **not** normative. The namespaces to which the prefixes refer are normative, however.

Table 1 — Namespace mappings

Prefix	Namespace URI	Description
`xsd`	http://www.w3.org/2001/XMLSchema	XML Schema namespace
`gml`	http://www.opengis.net/gml/3.2	GML 3.2.1
`crsnts`	http://www.opengis.net/CRS-NTS/1.0	CRS Name Type Specification

Note the `crsnts` namespace is only needed for dealing with Parameterized CRSs as described in Clause 10. It is not needed for handling CRS URIs, nor is it needed for handling GML CRS definitions.

6.2 BNF notation

BNF notation used is the same as in IETF RFC 2396 [3].

7 CRS definition and identifier

7.1 Overview

This Clause establishes requirements class *crs-nts*. This is the mandatory core conformance class of this specification.

When expressing the location of some object in space and time through coordinates, these coordinates need to have an unambiguous reference which usually is given by a Coordinate Reference System (CRS). Typically, however, it is inconvenient for both producer and consumer of such information to communicate the complete definition of the CRS used; rather, a unique identifier is preferred – even more so as the vast majority of applications relies on some community accepted standard CRSs.

The OWS Common standard [OGC 06-121r9] specifies that each specific OGC Web Service shall always reference a CRS by using a XML attribute or element with the type `anyURI`. Such an `anyURI` value can be used to reference a CRS whether or not the definition of that CRS is included in the same data transfer and whether or not can be accessed electronically from some server. Subclause 10.3.2 of OWS Common [OGC 06-121r9] specifies when and how to use URLs to reference a CRS or CRS-related object; subclause D.14 summarizes many of the requirements for referencing CRSs.

This OGC® Name Type Specification for Coordinate Reference Systems specification establishes a syntax for such `anyURI` items. In its basic form, a URI uniquely identifies a CRS (see Clause 7). Through a parametrization mechanism, the mere identification is extended beyond single CRS Identifiers to also express sets of CRSs, instantiations obtained from Parameterized CRSs, and ad-hoc combination of CRS components. This document does not specify where these different CRS URI mechanisms are allowed; for example, OGC specifications may allow Parameterized URIs in some places, and sometimes require Unique CRS URIs.

Example A typical situation occurs with the OGC Web Coverage Service (WCS) [OGC 09-110r4]. Its CRS Extension [OGC 11-053] allows incompletely specified CRS URIs in the Capabilities document, thereby enabling a compact presentation of the CRSs supported by a particular service. In a *GetCoverage* request, on the other hand, a unique CRS URI must be passed to the server to concretely specify the coordinate transformation to be applied.

7.3 Syntax

7.3.1 General CRS Identifier syntax

Requirement 1 req/query-format:
A CRS Identifier **shall** conform with the URL format as defined in IETF RFC 2396 [3]:

```
[ "http:" hier_part [ "?" query ] ] [ "#" fragment ]
```

Requirement 2 req/reserved-chars:
In a CRS Identifier, characters in values which are reserved in [OGC 09-048r3] **shall** be encoded in the percent notation defined in IETF RFC 2396 [3].

Example 1 A coverage document may contain a local CRS definition referred to by the `srsName` attribute: `srsName="#my_local_CRS"`

Example 2 Company ACME may offer CRS definitions (i.e., employ a CRS Identifier resolver) understanding CRS URLs like http://www.acme.com/def/this-is-EPSG-4326.

Example 3 The following URL resolves to a set of CRSs, namely all EPSG CRSs, in their current version, available in the OGC CRS Name Resolver: http://www.opengis.net/def/crs/EPSG/0/.

Note Generally, not all possible parameter values will identify some existing (or meaningful) CRS. The authoritative decision about validity of OGC CRS URIs is with the OGC CRS Name Resolver with service endpoint www.opengis.net/def.

Example 1 The OGC CRS Name Resolver returns a GML CRS definition of the WGS84 reference system when provided with URL http://www.opengis.net/def/crs/EPSG/0/4326.

Example 2 The OGC CRS Name Resolver returns a 404 error code when provided with URL http://www.opengis.net/def/crs/EPSG/0/this-is-not-a-valid EPSG-identifier.

7.3.2 GET/KVP format

Requirement 3 req/query-format:
A CRS Identifier in GET/KVP format **shall** be constructed as an http GET query, following the common URL format as defined in IETF RFC 2396 [3]:

```
"http:" hier_part  "?" par1 "=" val1 "&" par2 "=" val2 "&" …
```

where `par1`, `par2` etc. represent parameter names and `val1`, `val2`, etc. the value passed for the corresponding parameter as described in Table 2.

Note Like URIs in general [3], the domain part is case insensitive while the path and query part is case sensitive.

Admissible parameters for the query part (i.e., following the "?" character) in a given CRS or set of CRSs are those given by the service and by the individual CRS(s) interrogated, therefore individual. Table 2 lists CRS URI parameters reserved by OGC, together with their meanings. A particular CRS may define further parameters, as long as these do not conflict with the definitions of this standard.

Every parameter value can be instantiated only once, i.e., duplicate keys are not allowed.

Requirement 4 req/no-dupes:
A CRS Identifier **shall** not contain duplicate keys.

Example The following does not identify a CRS, but an (infinite) CRS set because the `CenterLatitude` parameter is not provided (cf. Annex A):
http://www.opengis.net/def/crs?authority=OGC&version=1.3&code=AUTO42003&UoM=m&CenterLongitude=-100

Note A CRS URI which does not specify authority, version, and code will result in a list of all CRSs matching this pattern; in particular, if none of these components is specified the resolver will return all CRS definitions known to this service. In case of a client side programming bug this would constitute an undesirable behaviour as the amount of data presumably will be substantial. A server, therefore, may implement internal limits in the amount of information delivered and respond with an exception upon transgression of this limit.

Table 2 — CRS URI query parameters

name	definition	data type
`authority`	The OGC-specified abbreviation for the authority organization that specified the referenced definition. As such, it identifies an authority recognized by the OGC. Example Among the currently recognized authorities are “OGC, “EPSG”, and “ISO”.	`NCName`
`version`	The version of the authority or code for the referenced definition. When the referenced definition does not have a version the string “0” (without quotes) shall be used.	`String`
`code`	Unique identifier of the referenced CRS definition, as specified by the referenced authority. This identifier shall be unique in combination with authority and version. Example “4326” is the EPSG code for WGS84.	`String`

Note The `version` format is sometimes of syntax “N.N.N” or “N.N”, where each “N” stands for an integer. No "v" or other version prefix is used by OGC. If no other version identification is provided by the authority, a four-digit year or other date can be used.

7.3.3 Path format

In the special case that the CRS identifier is one of

- Authority, but no version nor code;
- authority and version, but no code;
- authority and version and code,

the path variant, i.e., a RESTful syntax, can be used alternatively to the query syntax.

Requirement 5 req/path-format:
A CRS Identifier in path format, that is:

```
   ″http:″ hier_part ″/″ AUTHORITY ″/″ VERSION ″/″ CODE
```

shall be equivalent to

```
   ″http:″ hier_part ″?″ ″authority″ ″=″ AUTHORITY
                     ″&″ ″version″ ″=″ VERSION
                     ″&″ ″code″ ″=″ CODE
```

where the placeholders `AUTHORITY`, `VERSION`, and `CODE` denote admissible values for the corresponding lower-case variable parts defined in Table 2.

Example 1 The two URIs below are equivalent in identifying the WGS84 CRS:
http://www.opengis.net/def/crs/EPSG/0/4326
http://www.opengis.net/def/crs?authority=EPSG&version=0&code=4326

Example 2 The URI for the OGC CRS with code 4326 specified in the EPSG database [2] can be expressed as
?http://www.opengis.net/def/crs/EPSG/0/4326

Example 3 The URI for all OGC / EPSG CRSs in the EPSG database [2] can be expressed as
http://www.opengis.net/def/crs/EPSG

Example 4 A fictitious oarganization, ACME, might establish its own proprietary EPSG 4326 definition through
http://www.acme.com/def/this-is-EPSG-4326.

Note This syntax establishes backward compatibility to pre-existing OGC CRS URIs.

7.4 Semantics

The semantics of a CRS URI is given by the response obtained from the server receiving and resolving the URI. Depending on the completeness of the CRS URI, different cases are possible:

- If the CRS URI references a complete CRS definition by containing `authority`, `version`, and `code` parameters identifying a GML document whose root is in the substitution group of `gml:AbstractCRS`) then the response is this CRS definition.

- If in the CRS URI references a set of definitions by leaving out one or more of the `authority`, `version`, and `code` parts then the response is an unordered sequence of CRS definitions, with one definition for each value combination available on the server.

Requirement 6 req/representation-complete-crs:
The resource pertaining to a CRS URI which identifies a CRS whose root is in the substitution group of `gml:AbstractCRS` **shall** be the definition of the CRS identified.

Requirement 7 req/representation-crs-list:
The resource pertaining to a CRS URI referring to CRS definition **,** but with either `code` or `version` and `code` or `authority`, `version`, and `code` parameters missing in the URI **shall** be a `crnts:CRSList` containing exactly those CRS definitions stored in the server which match the CRS URI and where the missing parameters act as wildcards matching any value.

CRS and CRS list responses are expressed in GML.

Requirement 8 req/representation-gml-default:
By default, the representation delivered in response to a CRS Identifier **shall** be encoded in GML [OGC 07-036].

Note In future, further encodings beyond GML may be supported, such as WKT.

Example 1 A GML document describing the WGS84 reference system can be obtained through URL
http://www.opengis.net/def/crs/EPSG/0/4326

Example 2 The following URL does not provide a value for parameter `CenterLongitude` and, hence, will return the Parameterized CRS definition, which has as its document root a `crsnts:ParameterizedCRS` element:
http://www.opengis.net/def/crs?authority=OGC&version=1.3&code=AUTO42003&UoM=m

8 Axis identifiers

8.1 Overview

Coordinate system axes can be spatial (example: latitude, elevation), temporal, or none of both (example: pressure). While these concepts are uniquely defined, they frequently have several synonyms, for example, elevation is synonymous to bathymetry, altitude, height, and z. Likewise, CRS Axis URIs uniquely define axes for use in CRS definitions, but allow synonyms.

8.2 Syntax

Requirement 9 req/path-format:
A *CRS Axis Identifier* **shall** be a URL which has the common URL format as defined in IETF RFC 2396 [3]:

```
[ "http:" hier_part [ "?" query ] ] [ "#" fragment ]
```

Note Following URI construction rules [3], parameter keys in the `query` are case insensitive, while values are case sensitive.

Example 1 A coverage document may contain a local CRS definition containing a locally defined axis referred to by `#my_local_z_axis`.

Example 2 Company ACME may offer CRS definitions (i.e., employ a CRS Identifier resolver) understanding CRS URLs like http://www.acme.com/def/axis/this-is-latitude.

Example 3 The OGC definition of longitude is retrieved through this URL when passed to the OGC CRS Resolver: http://www.opengis.net/def/axis-name/EPSG/0/9902.

Requirement 10 req/reserved-chars:
In an Axis Identifier, characters in values which are reserved in [OGC 09-048r3] **shall** be encoded in the percent notation defined in IETF RFC 2396 [3].

8.3 Semantics

Requirement 11 req/semantics:
The resource identified by an Axis Identifier **shall** be the GML representation of the coordinate axis identified, as a GML document with root element `gml:CoordinateSystemAxis`.

Example Axis URL http://www.opengis.net/def/axis/elevation identifies this axis:

```
<gml:CoordinateSystemAxis gml:id="ogrcrs4" gml:uom="urn:ogc:def:uom:EPSG::9904">
  <gml:name>elevation</gml:name>
  <gml:axisID>
    <gml:name gml:codeSpace="urn:ogc:def:axis:EPSG::">9904</gml:name>
  </gml:axisID>
```



```
    <gml:axisAbbrev>z</gml:axisAbbrev>
    <gml:axisDirection>up</gml:axisDirection>
  </gml:CoordinateSystemAxis>
```

Axis definitions contain an element `gml:AxisAbbrev` which defines a short name for an axis. When this name is unambiguous (e.g., when the CRS under consideration is specified additionally) this axis abbreviation can be used in place of the full URI axis identifier.

Requirement 12 req/abbreviation:
In a context where the CRS on hand is defined unambiguously, using an identifier URI of one of the axes defined in said CRS **shall** be equivalent to any of the axis abbreviations defined in the CRS referenced or in some `<gml:AxisAbbrev>` element within the axis definition referenced by the axis URI.

Example The WGS84 CRS, and geographic CRSs in general, adopts the geodetic latitude axis, identified by http://www.opengis.net/def/axis/EPSG/0/106. This axis uses measures in degrees (e.g. referenced as `deg`, as by uom attribute http://www.opengis.net/def/uom/EPSG/0/9122). It is abbreviated as `Lat` whose name is stated as `Geodetic latitude` in the axis definition (http://www.opengis.net/def/axis-name/EPSG/0/9901). Thus, the envelope of a coverage in the geographic space can be described as follows:

```
<gmlcov:GridCoverage … gml:id="C0001">
    <gml:boundedBy>
        <gml:Envelope
          srsName="http://www.opengis.net/def/crs/EPSG/0/4326”
          axisLabels="Lat Long"
          uomLabels="deg deg"
          srsDimension="2">
            <gml:lowerCorner>1 1</gml:lowerCorner>
            <gml:upperCorner>3 10</gml:upperCorner>
        </gml:Envelope>
    </gml:boundedBy>
    …
</gmlcov:GridCoverage>
```

A CRS must not contain the same axis identification twice. Should a CRS need to contain two axis of the same kind – e.g., two temporal dimensions as used in climate simulation – then the axis abbreviation must be changed for one to be different.

Requirement 13 req/compatibility:
In a CRS Identifier, any given axis **shall** not occur more than once.

Example An axis renaming can be done by way of query parameters over a Parameterized CRS definition (cf. Clause 10) which targets the reference CRS. Assuming a (hypothetical, not existing) Parameterized CRS http://www.opengis.net/def/crs/myAuth/myVersion/my-EPSG-4267 has been created, enabling axes abbreviations' customization on the target CRS http://www.opengis.net/def/crs/EPSG/0/4267 (2D NAD27 CRS) via parameters called `lat-label` and `long-label`, then the following URI renames the axis abbreviation to `myPersonalLatitudeLabel` and `myPersonalLongitudeLabel`:

```
http://www.opengis.net/def/crs/myAuth/myVersion/my-EPSG-4267?lat-
label=myPersonalLatitudeLabel&long-label=myPersonalLongitudeLabel
```

An envelope GML description would then be:

```
<gmlcov:GridCoverage … gml:id="C0001">
    <gml:boundedBy>
        <gml:Envelope srsName="http://www.opengis.net/def/crs/
                                myAuth/myVersion/my-EPSG-4267?
                                lat-label=myPersonalLatitudeLabel&
                                long-label=myPersonalLongitudeLabel"
            axisLabels="myPersonalLatitudeLabel myPersonalLongitudeLabel"
            uomLabels="deg deg"
            srsDimension="2">
            <gml:lowerCorner>1 1</gml:lowerCorner>
            <gml:upperCorner>3 10</gml:upperCorner>
        </gml:Envelope>
    </gml:boundedBy>
      …
</gmlcov:GridCoverage>
```

9 Compound CRS

9.1 Overview

A *Compound CRS* is a CRS consisting of a non-repeating sequence of two or more single coordinate reference systems, none of which can itself be compound, as defined in OGC Abstract Topic 2 / ISO 19111-2 [OGC 08-015r2]. This way, Compound CRSs allow to establish higher-dimensional CRSs. This Clause describes a URI-based naming scheme for Compound CRSs.

9.2 Syntax

A special case of a CRS URI is when the parameter value provided itself is a URI, such as CRSs and coordinate axes.

A *Compound CRS URI* is a CRS Identifier with code `crs-compound`, together with further numbered parameters whose values are URIs themselves. A Compound CRS URI describes recombination of CRSs into a new CRS.

Requirement 14 req/format:
A CRS Identifier in Compound CRS URI format **shall** contain a path element `crs-compound`.

Requirement 15 req/input-crs-list:
A CRS Identifier in Compound CRS URI format **shall** contain $n>1$ query parameters with keys "1" through "n" (in decimal notation) whose corresponding values are CRS Identifiers.

Requirement 16 req/predefined-crs-only:
A CRS Identifier in Compound CRS URI format **shall**, in its query parameters, only contain CRS Identifiers for complete CRSs, i.e., CRSs whose root element is in the substitution group of `gml:AbstractCRS`.

Requirement 17 req/no-duplicates:
A CRS Identifier in Compound CRS URI format **shall** not contain duplicate CRS Identifiers in the numbered query parameters.

Note 1 A CRS Identifier, regardless of whether expressed in query or path format, may contain additional query parameters, as long as no parameter name conflicts occur.

Note 2 There does not need to pre-exist any well-known CRS definition matching this new CRS.

Example Obtaining pressure/time slices from a 4D latitude/longitude/time/pressure atmospheric data cube requires a pressure/time CRS, for which no CRS standard exists. A Compound CRS allows to express a CRS identifier for this situation.

Requirement 18 req/unique-axes:
In a Compound CRS URI, all axes in the CRSs listed **shall** be pairwise different.

Note Axes can be the same (e.g., Northing) while their identifiers are distinct.

Example 1 To define a 3D spatio-temporal CRS for satellite image time series, with axis order latitude-longitude-time, the WGS84 CRS may be combined with a temporal CRS, such as *ansi-date*[2]:

http://www.opengis.net/def/crs-compound?
1=http://www.opengis.net/def/crs/EPSG/0/4326
&2=http://www.opengis.net/def/crs/OGC/0/ansi-date

Example 2 The URI for combining the CRSs EPSG 4269 (NAD83) and EPSG 5713 (Canadian Geodetic Vertical Datum of 1928), adding in height, is:

http://www.opengis.net/def/crs-compound?
1=http://www.opengis.net/def/crs/EPSG/0/4269
&2=http://www.opengis.net/def/crs/EPSG/0/5713

Requirement 19 req/axis-sequence:
In the CRS definition resulting from resolving a Compound CRS, axes **shall** appear in the order (i) of the query parameters listed in the URI and (ii) of the axis order within each CRS listed.

Example In a request like

http://www.opengis.net/def/crs-compound?1=crs1&2=crs2

where crs1 has axis order (a,b) and crs2 has axis order (c,d), the resulting CRS will have an axis order of (a,b,c,d).

10 Parameterized CRS

10.1 Overview

A *Parameterized CRS* is a CRS which contains variables.

10.2 Syntax

A Parameterized CRS definition consists of a reference to a CRS definition plus a list of parameter names (i.e., variables) and optional default values. A Parameterized CRS URI can contain a query part with a list of key/value pairs where values are assigned to variables defined in the Parameterized CRS.

[2] This CRS is not yet adopted by OGC; hence, it may end up having a different name.

10.3 Semantics

The semantics of a Parameterized CRS URI is given by the response obtained from the server receiving and resolving the URI. If a CRS URI references a Parameterized CRS definition (i.e., a GML document whose root is of type `crsnts:ParameterizedCRS`) then the resolver response is obtained from the Parameterized CRS definition by substituting all formal parameters by their values. These values can be provided either in the CRS URI or can be satisfied via default values in the Parameterized CRS definition itself.

- If, after this substitution, some unsatisfied parameters remain then the response is of type `crsnts:ParameterizedCRS` and contains only those reference `crsnts:Parameters` which remain unsatisfied.

- If, after this substitution, no unsatisfied parameters remain then the response is in the substitution group of `gml:AbstractCRS`.

Requirement 20 req/representation-parameterized-incomplete:
The resource pertaining to a CRS URI which identifies a CRS whose root is in the substitution group of `crsnts:ParameterizedCRS` and where not all formal parameters of this Parameterized CRS definition are provided (either in the CRS URI or satisfied via default values in the Parameterized CRS definition) **shall** be the `crsnts:ParameterizedCRS` with all parameters provided are removed from the `crsnts:parameters` list and their values properly substituted in the `crsnts:targetReferenceSystem`.

Requirement 21 req/representation-parameterized-complete:
The resource pertaining to a CRS URI which identifies a CRS whose root is in the substitution group of `crsnts:ParameterizedCRS` and where all formal parameters of this Parameterized CRS definition are provided (either in the CRS URI or satisfied via default values in the Parameterized CRS definition)
shall be the `gml:AbstractCRS` definition referenced in the `crsnts:targetReferenceSystem` of the CRS identified, with all parameters properly substituted.

Example The following is a sample Parameterized CRS definition expressed in GML:

```
<ParameterizedCRS
    xmlns="http://www.opengis.net/crs-nts/1.0"
    xmlns:gml="http://www.opengis.net/gml/3.2"
    xmlns:xlink="http://www.w3.org/1999/xlink"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    gml:id="parameterized-crs-4326">
    <gml:identifier codeSpace="JUB">
        http://www.opengis.net/def/crs/AUTO/1.3/42001
    </gml:identifier>
    <gml:scope>unknown</gml:scope>
    <parameters>
        <parameter name="lon"/>
        <parameter name="lat">
            <value>0.0</value>
        </parameter>
        <parameter name="zone">
            <value>min( floor( (${lon} + 180.0) / 6.0 ) + 1, 60 )</value>
        </parameter>
```



```
        <parameter name="central_meridian">
            <value>-183.0 + ${zone} * 6.0</value>
            <target>//greenwichLongitude</target>
        </parameter>
        <parameter name="false_northing">
            <value>(${lat} >= 0.0) ? 0.0 : 10000000.0</value>
            <target>//falseNorthing</target>
        </parameter>
    </parameters>
    <targetReferenceSystem
        xlink:href="http://www.opengis.net/def/crs/EPSG/0/4326"/>
</ParameterizedCRS>
```

Bibliography

- none here -

Annex A
(non-normative)

OGC CRS Name Resolver

This Annex describes the OGC CRS Name Resolver for CRS URIs as defined in this specification. The OGC CRS Name Resolver delivers definitions identified by CRS and Axis URIs, provides Parameterized CRS definitions, and builds CRS specifications through composition and concatenation.

A.1 Implementation and availability

The concepts established in this document describe the OGC CRS resolver. This CRS Name Resolver, SECORE (Semantic Cordinate Reference System Resolver),is implemented in Java using the BaseX embedded XML database system (www.basex.org). The source code of SECORE is provided and maintained by Jacobs University and rasdaman GmbH as open-source Java code published under GNU LGPL as part of the rasdaman software. In the rasdaman source tree available from www.rasdaman.org it can be found in subdirectory `applications/secore/`.

A.2 Service elements

The OGC CRS Name Resolver accepts Axis and CRS Identifiers as input URLs. Further, it accepts general XQuery requests on its CRS database.

A.3 Service URI

The OGC CRS Name Resolver is accessible at the following service endpoint:

- http://www.opengis.net/def/axis for Axis Identifier URLs
- http://www.opengis.net/def/crs for CRS Identifier URLs and Parameterized CRS URLs
- http://www.opengis.net/def/crs-compound for Compound CRS URLs
- http://www.opengis.net/def/crs-equals for semantic CRS URL comparison, see A.8
- http://www.opengis.net/def/crs-query for general XQuery requests, see A.9

A.4 Service syntax

The OGC CRS Name Resolver accepts queries in the syntax defined in this Name Type Specification.

A.5 URI Resolution

Resolution of a CRS URI results in either an Axis definition, a CRS definition, a Parameterized CRS definition, a CRS set, or an exception.

Parameters in the URI query must all match; non-matching parameters will raise an exception.

A.6 Parameterized CRS definitions

A.6.1 Overview

CRS definitions can be parameterized with one or more named parameters which are declared explicitly in the corresponding definition. As such, they describe (possibly infinite) sets of concrete CRSs.

Note This term "parameterized" should not be mixed up with the term "parametric" in OGC Abstract Topic 2 / ISO 19111-2 [10] which has a significantly different meaning.

Parameters can be resolved through values provided in the CRS URI, or through defaults defined in the CRS definition. Additionally, expressions ("formulae") can be associated with a Parameterized CRS which evaluate to values when instantiated with parameter values. All values, whether instantiated in a URL request or coming from a default or a formula, can be substituted in one or several places in the concrete CRS definition associated with the Parameterized CRS.

Example The following URI defines the Auto Orthographic CRS 42003 specified in Subclauses 6.7.3.4 and B.9 of WMS 1.3 [4] for "meter" as unit of measure and centered at 100° West longitude and 45° North latitude:
http://www.opengis.net/def/crs?authority=OGC&version=1.3&code=AUTO42003&
UoM=m&CenterLongitude=-100&CenterLatitude=45

Note Additional examples of not-completely-specified objects are specified in Subclauses B.7, B.8, B.10, and B.11 of WMS 1.3 [4], and in Subclauses 10.1 through 10.3 of [OGC 05-096r1] .

A.6.2 Structure

Formally, a *Parameterized CRS* is a GML document with root `crsnts:ParameterizedCRS`. It contains an element `crsnts: targetReferenceSystem` referring to some instantiatable subtype of `gml:AbstractCRS`, together with a list of formal parameters.

Parameters are `crsnts:Parameter` elements listed in the `crsnts:Parameters` section. A formal parameter consists of

- a `name` which is locally unique within the `Parameters` element,
- an optional `crsnts:value` element holding a default value consisting of either a constant or a formula (see below for its syntax), and
- an optional `crsnts:target` containing an XPath expression indicating one or more substitution points relative to the `targetReferenceSystem` subnode.

When the `crsnts:value` element contains a well-formed formula then this formula shall adhere to the JSR scripting syntax as specified in JSR-233 [5]. The formula shall contain only references to parameter names defined in the same Parameterized CRS. No (direct or indirect) recursive references across formulae are allowed. Variable names referenced are enclosed in `${` and `}`.

Note In particular, a formula cannot have its own parameter name as a free parameter.

The target expression in `crsnts:target` indicates the places where, during request evaluation, the resulting parameter (obtained from URL input, or formula evaluation, or by using the default) gets applied to the CRS definition, assuming `crsnts:targetReferenceSystem` as the relative document root for XPath evaluation. The XPath item indicates one or a set of substitution points relative to the CRS subnode.

Example The following XML snippet defines a geodetic Parameterized CRS with formal parameter `x` substituting parameter values in all (fictitious) `axisName` elements appearing the `gml:GeodeticCRS` root of the CRS definition:

```
<crsnts:ParameterizedCRS>
  <gml:identifier>
  </gml:identifier>
  <gml:scope>…</gml:scope>
  <crsnts:parameters>
    <crsnts:parameter name=”lon” >
      <crsnts:value>90</crsnts:value>
      <crsnts:target>//longitude | //Longitude</crsnts:target>
    </crsnts:parameter>
    <crsnts:parameter name="zone">
      <crsnts:target>//greenwichLongitude</crsnts:target>
      <crsnts:value>
        min(floor((${lon} + 180.0) / 6.0) + 1,60)
      </crsnts:value>
    </crsnts:parameter>
  </crsnts:parameters>
  <crsnts:targetReferenceSystem
   xlink:href="http://www.opengis.net/def/crs/EPSG/0/4326"/>
</crsnts:ParameterizedCRS>
```

A.6.3 Resolution

The result of a URI request against a Parameterized CRS depends on the degree of parameter matching, it is a GML document with its root being an instantiatable subtype of either `gml:AbstractCRS` or `crsnts:ParameterizedCRS`. The response is:

- In case all formal parameters in the Parameterized CRS addressed are resolved the result is a `gml:AbstractCRS`.

  Example Assuming that the name of the above Parameterized CRS example is `my-own-crs`, a possible instantiation of this Parameterized CRS to a concrete CRS Identifier is
  http://www.opengis.net/def/crs/my-own-crs?lon=47.6

  The response to this instantiation is

  ```
  <gml:GeodeticCRS>
    …
  <gml:GeodeticCRS>
  ```

- In case not all parameters are resolved, the resolved parameters are removed, and the non-matched parameters remain in the template. The result is a `crsnts:ParameterizedCRS`.

Example Assuming the same example as above, the Parameterized CRS itself can be obtained through http://www.opengis.net/def/crs/my-own-crs

The response to this request is

```
<crsnts:ParameterizedCRS>
  <gml:identifier>
  </gml:identifier>
  <gml:scope>…</gml:scope>
  <crsnts:parameters>
    …
  </crsnts:parameters>
  <crsnts:targetReferenceSystem xlink:href="…"/>
</crsnts:ParameterizedCRS>
```

The corresponding XML Schema is available from http://schemas.opengis.net/crs-nts.

A.7 Flattening of CRS definitions

As opposed to classic CRSs, the concepts introduced here allow nested and parameterized definitions of CRSs. When resolving a CRS URI, tools and human users sometimes may want to obtain the high-level definition indicating how a CRS is composed form other CRSs, and sometimes may want to get an expanded view with all references to other CRSs resolved. An optional parameter in URIs allows obtaining CRS definitions at selectable degrees of resolution.

A CRS URI can be appended with an optional query parameter, `expand`, which allows to control how far `xlinks` in the document returned are resolved. This parameter can have the following values:

- `none`: do not follow and resolve any `xlink` element in the CRS definition addressed.
- `all`: resolve all `xlink` elements in the CRS definition addressed. This is the default.
- `n` where `n` is an integer number greater than zero: resolve `xlinks` up to `n` nesting levels.

Example 1 `...&expand=none` – during resolution, do not resolve any `xlinks`.

Example 2 `...&expand=all` – during resolution, resolve all `xlinks`.

Example 3 `...&expand=2` – during resolution, follow up to two `xlinks` in sequence.

A.8 CRS equality

It is possible that one and the same CRS, axis, etc. is identified by a number of syntactically different URLs, and it is not straightforward for applications to decide about equivalence of two given URIs. To remedy this, a comparison predicate is available in the resolver. A request containing two URLs listed as GET/KVP parameters with names `1` and `2`, respectively, will result in a response of `true` if and only if both URLs resolve to definitions equivalent on the semantic level of XML, and `false` otherwise; the response is embedded in an XML document.

Example Comparing EPSG codes 4327 and 4326 can be done with this URL:

```
http://www.opengis.net/def/crs-equals?
      1=http://www.opengis.net/def/crs/EPSG/0/4327
     &2=http://www.opengis.net/def/crs/EPSG/0/4326
```

The response will look like this:

```
<crsnts:comparisonResult
  xmlns='http://www.opengis.net/crs-nts/1.0'>
  <crsnts:equal>false</crsnts:equal>
  <crsnts:reason>
    <![CDATA[ ...description text... ]]>
  </crsnts:reason>
</crsnts:comparisonResult>
```

Note that the text provided in `crsnts:reason` is not standardized, but implementation dependent.

A.9 XQuery

An XQuery GET or POST request sent to URL http://www.opengis.net/def/crs-query will result in a document obtained from evaluating the XQuery request according to the XQuery standard [6].

A.10 Exceptions

Exceptions shall be thrown in the situations described in Table 3, with the http code and values indicated there.

Table 3 — CRS resolver exception codes

`exceptionCode` value	HTTP code	Meaning of exception code	`locator` value
NO_SUCH_CRS	404	CRS not available on this server.	First offending range field name in the parameter list
CANNOT_ENUMERATE_CRS_SET	404	Query resolves to an infinite number of CRSs.	none
DUPLICATE_KEY	404	Query contains duplicate keys in the query part.	First duplicate
NO_SUCH_KEY	404	Query contains a non-matching key (i.e., a key which has no meaning for the resolver) in the query part.	First non-matching key
XQUERY_ERROR	404	XQuery request passed is not well-formed or cannot be evaluated against database contents.	none

NO_SUCH_EXPAND_ VALUE	404	An `expand` query parameter has been provided with an illegal value.	`expand` parameter value