Open Geospatial Consortium

Submission Date: <yyyy-mm-dd>

Approval Date: <yyyy-mm-dd>

Publication Date: <yyyy-mm-dd>

External identifier of this OGC® document: <http://www.opengis.net/doc/[{doc-type/}]{standard}/{m.n}>

Internal reference number of this OGC® document: YY-nnnrx

Version: n.n.n

Category: OGC® Implementation Specification

Editor: Pekka Latvala

OGC Table Joining Service 2.0 - Part 1: Core

Copyright notice

Warning

This document is not an OGC Standard. This document is distributed for review and comment. This document is subject to change without notice and may not be referred to as an OGC Standard.

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.

Document type:	OGC® Standard
Document subtype:	Interface
Document stage:	Draft
Document language:	English

Contents

i. Abstract

This document is a specification of the core module of the Table Joining Service (TJS) 2.0 standard. The TJS allows attribute datasets that don't contain any spatial information to be joined with spatial datasets. The joining between the datasets cam be executed via shared identifiers.

The TJS specification is a multi-part document that can be extended by specifying additional modules to the core. Some potential modules are the data maintenance module for handling inserts, updates and deletes and the identifier mapping module.

The document specifies the core functionalities that all TJS implementations must support. The core functionalities are executed via HTTP GET queries and they are restricted to displaying the metadata and data values from different data sets, displaying service's join output capabilities and executing the joins between the attribute datasets and the spatial datasets. Table 1 contains an overview of the operations that are specified in the TJS core module.

Table 1 Overview of operations in the TJS core module

Path	**HTTP Method**	**Description**
/attributedatasets	GET	Returns metadata on all attribute datasets available on the server
/attributedatasets/{attributedatasetid}	GET	Returns detailed metadata on a specific attribute dataset
/attributedatasets/{attributedatasetid}/data	GET	Returns the data values of a specific attribute dataset
/spatialdatasets	GET	Returns metadata on all spatial datasets available on the server
/spatialdatasets/{spatialdatasetid}	GET	Returns the metadata on a specific spatial dataset
/spatialdatasets/{spatialdatasetid}/key	GET	Returns the key values of a specific spatial dataset
/joinabilities	GET	Returns metadata on server's join abilities
/joindata/{spatialdatasetid}?attributedataseturl	GET	Executes a join between a specific spatial dataset and attribute dataset located in the

		address of attributedataseturl query parameter

ii. Keywords

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

ogcdoc, OGC document, table joining service, TJS, API, openapi

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.

iv. Submitting organizations

The following organizations submitted this Document to the Open Geospatial Consortium (OGC):

- National Land Survey of Finland

v. Submitters

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

Name	Affiliation
Pekka Latvala	National Land Survey of Finland

1. Scope

The Table Joining Service 2.0 core nodule specifies a RESTful service interface that contains operations for displaying metadata and data on attribute datasets, metadata and key values on spatial datasets, metadata on server’s join capabilities and an operation for executing joins between attribute datasets and spatial datasets.

The core module doesn’t contain any functionalities for dataset maintenance. These functionalities can be defined in the possible additional modules. Other useful functionalities for additional modules include identifier mapping module and a module for uploading and on-the-fly joining of various data formats, such as CSV files.

2. Conformance

This document defines 1 requirement / conformance class, “core”.

The core conformance class specifies the requirements that all TJS implementations must support.

Requirements are considered for 1 standardization target type:

- Web services

The common response format for all operations defined in the core conformance class is the JavaScript Object Notation (JSON) format. The core module does not specify any output formats for the joined datasets and the service implementations can support any formats they choose.

The core module does not specify any mandatory formats for the API definition. Some options that can be used for the API definition are different versions of the OpenAPI specification.

Conformance with this standard shall be checked using all the relevant tests specified in Annex A (normative) of this document. The framework, concepts, and methodology for testing, and the criteria to be achieved to claim conformance are specified in the OGC Compliance Testing Policies and Procedures and the OGC Compliance Testing web site1.

In order to conform to this OGC™ interface standard, a software implementation shall choose to implement:

a) Any one of the conformance levels specified in Annex B (normative).

b) Any one of the Distributed Computing Platform profiles specified in Annexes TBD through TBD (normative).

All requirements-classes and conformance-classes described in this document are owned by the standard(s) identified.

[1] www.opengeospatial.org/cite

3. References

The following normative documents contain provisions that, through reference in this text, constitute provisions of this document. 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.

<Insert References here. If there are no references, state "There are no normative references".>

4. Terms and Definitions

This document uses the terms defined in Sub-clause 5.3 of [OGC 06-121r8], 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 standard.

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

4.1
Attribute dataset
Definition…

4.2
Spatial dataset
Definition…

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 specification are denoted by the URI

```
http://www.opengis.net/spec/{standard}/{m.n}
```

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

6. Requirement Class “core”

6.1 Overview

The requirement class “core” contains 8 operations that all TJS services must implement. All of these operations are executed as HTTP GET queries. The common output format for all operation responses is JSON that the servers SHALL support. Servers may support also any other output formats they choose.

6.2 Operation /attributedatasets

The HTTP GET operation at path **/attributedatasets** returns a list containing basic metadata on all attribute datasets that are available on the server.

6.2.1 Request

Req 1	/req/core/attributedatasets-op The server SHALL support the HTTP GET operation at the path /attributedatasets

6.2.2 Response

Req 2	/req/core/attributedatasets-success A successful execution of the operation SHALL be reported as a response with a HTTP status code 200 For successful operation, the server SHALL return a list containing basic metadata of all attribute datasets available on the server
	/req/core/attributedatasets-failure If attribute datasets are not found on the server, it SHALL be reported as a response with a HTTP status code 404
	/req/core/attributedatasets-error If there is an error in the server during the processing of the request, it SHALL be reported as a response with a HTTP status code 500

6.2.3 Schema for the /attributedatasets response

```
AttributeDatasetsObject
type	"object"
properties
	relatedSpatialDataset
		$ref	"#/definitions/RelatedSpatialDatasetObject"
	attributeDatasetId
```

```
                type    "integer"
                format  "int32"
        attributeDatasetURI
                type    "string"
        relatedSpatialDatasetURI
                type    "string"
        organization
                type    "string"
        title
                type    "string"
        attributeDatasetAbstract
                type    "string"
        referenceDate
                type    "string"
        startDate
                type    "string"
        version
                type    "string"
        documentation
                type    "string"



RelatedSpatialDatasetObject
type    "object"
properties
        spatialDatasetId
                type    "integer"
                format  "int32"
        spatialDatasetURI
                type    "string"
        organization
                type    "string"
        title
                type    "string"
        metadataAbstract
                type    "string"
        referenceDate
                type    "string"
        startDate
                type    "string"
        version
                type    "string"
        documentation
                type    "string"
        spatialDatasetKeyName
                type    "string"
        spatialDatasetKeyType
                type    "string"
        spatialDatasetKeyLength
                type    "string"
        spatialDatasetKeyDecimals
                type    "string"
        bboxNorth
                type    "string"
        bboxSouth
                type    "string"
        bboxEast
                type    "string"
        bboxWest
                type    "string"
        spatialDatasetLink
                type    "string"
```

6.3 Operation /attributedatasets/{attributedatasetid}

The HTTP GET operation at path **/attributedatasets/{attributedatasetid}** returns detailed metadata on a specific attribute dataset specified with the path parameter **attributedatasetid.**

Attribute datasets may contain 4 types of attributes:

- Nominal
  - Nominal attributes have names as attribute values (i.e. Helsinki)
- Ordinal
  - Ordinal attributes have values that contain ranking information (i.e. low, medium and high categories)
- Count
  - Count attributes are numeric and their values are cumulative total for a region (i.e. population on a region)
- Measure
  - Measure attributes are numeric and their values are measurements on a region (i.e. rainfall on a region)

6.3.1 Request

Req 3	/req/core/req-attributedatasets-attributedatasetid -op The server SHALL support the HTTP GET operation at the path /attributedatasets/{attributedatasetid}

6.3.2 Response

Req 4	/req/core/req-attributedatasets-attributedatasetid-success A successful execution of the operation SHALL be reported as a response with a HTTP status code 200 For successful operation, the server SHALL return the metadata on a specific attribute dataset available on the server
	/req/core/req-attributedatasets-attributedatasetid-failure If the specific attribute dataset is not found on the server, it SHALL be reported as a response with a HTTP status code 404
	/req/core/req-attributedatasets-attributedatasetid-error If there is an error in the server during the processing of the request, it SHALL be reported as a response with a HTTP status code 500

6.3.3 Schema for the /attributedatasets/{attributedatasetid}response

```
AttributeDatasetObject
type    "object"
properties
        relatedSpatialDataset
                $ref    "#/definitions/RelatedSpatialDatasetObject"
        attributeDatasetId
                type    "integer"
                format  "int32"
        attributeDatasetURI
                type    "string"
        relatedSpatialDatasetURI
                type    "string"
        organization
                type    "string"
        title
                type    "string"
        attributeDatasetAbstract
                type    "string"
        referenceDate
                type    "string"
        startDate
                type    "string"
        version
                type    "string"
        documentation
                type    "string"
        columnset
                $ref    "#/definitions/ColumnsetObject"



RelatedSpatialDatasetObject
type    "object"
properties
        spatialDatasetId
                type    "integer"
                format  "int32"
        spatialDatasetURI
                type    "string"
        organization
                type    "string"
        title
                type    "string"
        metadataAbstract
                type    "string"
        referenceDate
                type    "string"
        startDate
                type    "string"
        version
                type    "string"
        documentation
                type    "string"
        spatialDatasetKeyName
                type    "string"
        spatialDatasetKeyType
                type    "string"
        spatialDatasetKeyLength
                type    "string"
        spatialDatasetKeyDecimals
                type    "string"
        bboxNorth
                type    "string"
        bboxSouth
                type    "string"
        bboxEast
                type    "string"
        bboxWest
                type    "string"
        spatialDatasetLink
                type    "string"
```

```
ColumnsetObject
type    "object"
properties
        frameworkKey
                $ref    "#/definitions/ColumnsetFrameworkKeyObject"
        attributes
                type    "array"
                items
                        $ref    "#/definitions/ColumnsetAttributeColumnObject"



ColumnsetFrameworkKeyObject
type    "object"
properties
        complete
                type    "boolean"
                default false
        relationship
                type    "string"
        type
                type    "string"
        name
                type    "string"



ColumnsetAttributeColumnObject
type    "object"
properties
        type
                type    "string"
        name
                type    "string"
        purpose
                type    "string"
        title
                type    "string"
        columnAbstract
                type    "string"
        documentation
                type    "string"
        count
                $ref    "#/definitions/ColumnsetAttributeColumnValuesCountObject"
        nominal
        $ref    "#/definitions/ColumnsetAttributeColumnValuesNominalObject"



ColumnsetAttributeColumnValuesCountObject
type    "object"
properties
    exceptions
        $ref
    "#/definitions/ColumnsetAttributeColumnValuesCountExceptionsObject"
    uom
        $ref
    "#/definitions/ColumnsetAttributeColumnValuesCountUOMObject"
    uncertainty
        $ref
    "#/definitions/ColumnsetAttributeColumnValuesCountUncertaintyObject"



ColumnsetAttributeColumnValuesCountExceptionsObject
type    "object"
properties
    identifier
        type    "string"
    title
        type    "string"
    exceptionAbstract
        type    "string"
    documentation
        type    "string"
    color
        type    "string"



ColumnsetAttributeColumnValuesCountUOMObject
type    "object"
properties
    reference
        type    "string"
    shortForm
        type    "string"
    longForm
        type    "string"
```

```
ColumnsetAttributeColumnValuesCountUncertaintyObject
type    "object"
properties
        gaussian
                type    "string"
```

```
ColumnsetAttributeColumnValuesNominalObject
type    "object"
properties
        classes
                $ref
        "#/definitions/ColumnsetAttributeColumnValuesNominalClassesObject"
        exceptions
                $ref
        "#/definitions/ColumnsetAttributeColumnValuesCountExceptionsObject"
```

```
ColumnsetAttributeColumnValuesNominalClassesObject
type    "object"
properties
        title
                type    "string"
        abstract_
                type    "string"
        documentation
                type    "string"
        value
                $ref
        "#/definitions/ColumnsetAttributeColumnValuesNominalClassesValueObject"
```

```
ColumnsetAttributeColumnValuesCountExceptionsObject
type    "object"
properties
        identifier
                type    "string"
        title
                type    "string"
        exceptionAbstract
                type    "string"
        documentation
```

```
                type     "string"
        color
                type     "string"



ColumnsetAttributeColumnValuesNominalClassesValueObject
type     "object"
properties
        identifier
                type     "string"
        title
                type     "string"
        abstract_
                type     "string"
        documentation
                type     "string"
        color
                type     "string"
```

6.4 Operation /attributedatasets/{attributedatasetid}/data

The HTTP GET operation at path **/attributedatasets/{attributedatasetid}/data** returns the data values of a specific attribute dataset specified with the path parameter attributedatasetid.

The operation MAY contain a query parameter **attributes** that can be used for selecting specific attributes for the response. If the attributes parameter is not used or available, the server SHALL return all attributes that are related to the particular attribute dataset.

6.4.1 Request

<table>
<tr>
<td><b>Req 5</b></td>
<td>/req/core/req-attributedatasets-attributedatasetid-data-op

The server SHALL support the HTTP GET operation at the path /attributedatasets/{attributedatasetid}/data

<b>Query parameters:</b>
<table>
<tr><td>Name</td><td>Description</td><td>Type and values</td><td>Required</td></tr>
<tr><td>attributes</td><td>The names of the attributes that are requested from the</td><td>String type. The value contains the names of the requested attributes in a comma-</td><td>Optional</td></tr>
</table>
</td>
</tr>
</table>

		attribute dataset	delimited format	

6.4.2 Response

Req 6	/req/core/req-attributedatasets-attributedatasetid-data-success A successful execution of the operation SHALL be reported as a response with a HTTP status code 200 For successful operation, the server SHALL return a data of a specific attribute dataset available on the server
	/req/core/req-attributedatasets-attributedatasetid-data-failure If the attribute dataset data are not found on the server, it SHALL be reported as a response with a HTTP status code 404
	/req/core/req-attributedatasets-attributedatasetid-data-error If there is an error in the server during the processing of the request, it SHALL be reported as a response with a HTTP status code 500

6.4.3 Schema for the /attributedatasets/{attributedatasetid}}/data response

```
RowsetObject
type    "object"
properties
        datasetUri
                type    "string"
        attributes
                type    "array"
                items
                        $ref    "#/definitions/RowsetAttributeObject"
        rowset
                type    "array"
                items
                        $ref    "#/definitions/RowsetRowObject"



RowsetAttributeObject
type    "object"
properties
        attributeName
                type    "string"
```

```
RowsetRowObject
type     "object"
properties
         k
                  type     "string"
         v
                  type     "array"
                  items
                           type     "string"
```

6.5 Operation /spatialdatasets

The HTTP GET operation at path **/spatialdatasets** returns a list containing metadata on all spatial datasets that are available on the server.

6.5.1 Request

Req 7	/req/core/req-spatialdatasets-op The server SHALL support the HTTP GET operation at the path /spatialdatasets

6.5.2 Response

Req 8	/req/core/req-spatialdatasets-success A successful execution of the operation SHALL be reported as a response with a HTTP status code 200 The server SHALL return a metadata on all spatial dataset available in the server
	/req/core/req-spatialdatasets-failure If the spatial datasets are not found on the server, it SHALL be reported as a response with a HTTP status code 404
	/req/core/req-spatialdatasets-error If there is an error in the server during the processing of the request, it SHALL be reported as a response with a HTTP status code 500

6.5.3 Schema for the /spatialdatasets response

```
SpatialDatasetObject
type    "object"
properties
        spatialDatasetId
                type    "integer"
                format  "int32"
        spatialDatasetURI
                type    "string"
        organization
                type    "string"
        title
                type    "string"
        metadataAbstract
                type    "string"
        referenceDate
                type    "string"
        startDate
                type    "string"
        version
                type    "string"
        documentation
                type    "string"
        spatialDatasetKeyName
                type    "string"
        spatialDatasetKeyType
                type    "string"
        spatialDatasetKeyLength
                type    "string"
        spatialDatasetKeyDecimals
                type    "string"
        bboxNorth
                type    "string"
        bboxSouth
                type    "string"
        bboxEast
                type    "string"
        bboxWest
                type    "string"
```

6.6 Operation /spatialdatasets/{spatialdatasetid}

The HTTP GET operation at path **/spatialdatasets/{spatialdatasetid}** returns metadata on a specific spatial dataset specified with the path parameter **spatialdatasetid**.

6.6.1 Request

Req 9	/req/core/req-spatialdatasets-spatialdatasetid-op The server SHALL support the HTTP GET operation at the path /spatialdatasets/{spatialdatasetid}

6.6.2 Response

Req 10	/req/core/req-spatialdatasets-spatialdatasetid-success A successful execution of the operation SHALL be reported as a response with a HTTP status code 200 The server SHALL return metadata on a specific spatial dataset available in the server
	/req/core/req-spatialdatasets-spatialdatasetid-failure If the spatial dataset is not found on the server, it SHALL be reported as a response with a HTTP status code 404
	/req/core/req-spatialdatasets-spatialdatasetid-error If there is an error in the server during the processing of the request, it SHALL be reported as a response with a HTTP status code 500

6.6.3 Schema for the /spatialdatasets/{spatialdatasetid} response

```
SpatialDatasetObject
type     "object"
properties
         spatialDatasetId
                  type     "integer"
                  format  "int32"
         spatialDatasetURI
                  type     "string"
         organization
                  type     "string"
         title
                  type     "string"
         metadataAbstract
                  type     "string"
         referenceDate
                  type     "string"
         startDate
```

```
        type    "string"
version
        type    "string"
documentation
        type    "string"
spatialDatasetKeyName
        type    "string"
spatialDatasetKeyType
        type    "string"
spatialDatasetKeyLength
        type    "string"
spatialDatasetKeyDecimals
        type    "string"
bboxNorth
        type    "string"
bboxSouth
        type    "string"
bboxEast
        type    "string"
bboxWest
        type    "string"
```

6.7 Operation /spatialdatasets/{spatialdatasetid}/key

The HTTP GET operation at path **/spatialdatasets/{spatialdatasetid}/key** returns a list of key values of a specific spatial dataset.

6.7.1 Request

Req 11	/req/core/req-spatialdatasets-spatialdatasetid-key-op The server SHALL support the HTTP GET operation at the path /spatialdatasets/{spatialdatasetid}/key

6.7.2 Response

Req 12	/req/core/req-spatialdatasets-spatialdatasetid-key-success A successful execution of the operation SHALL be reported as a response with a HTTP status code 200 The server SHALL return the key values of a specific spatial dataset.
	/req/core/req-spatialdatasets-spatialdatasetid-key-failure If the spatial dataset key values are not found on the server, it SHALL

	be reported as a response with a HTTP status code 404
	/req/core/req-spatialdatasets-spatialdatasetid-key-error If there is an error in the server during the processing of the request, it SHALL be reported as a response with a HTTP status code 500

6.7.3 Schema for the /spatialdatasets/{spatialdatasetid}/key response

```
SpatialDatasetKeyObject
type	"object"
properties
	key
		type	"string"
	title
		type	"string"
```

6.8 Operation /joinabilities

The HTTP GET operation at path /joinabilities returns a list of output formats that are supported by the server in the data join.

6.8.1 Request

Req 13	/req/core/req-joinabilities-op The server SHALL support the HTTP GET operation at the path /joinabilities

6.8.2 Response

Req 14	/req/core/req-joinabilities-success A successful execution of the operation SHALL be reported as a response with a HTTP status code 200 The server SHALL return metadata on the output formats that are supported by the server in the data join
	/req/core/req-joinabilities-failure If the join abilities metadata are not found on the server, it SHALL be reported as a response with a HTTP status code 404

	/req/core/req-joinabilities-error If there is an error in the server during the processing of the request, it SHALL be reported as a response with a HTTP status code 500

6.8.3 Schema for the /joinabilities response

```
OutputMechanismObject
type     "object"
properties
         identifier
                  type     "string"
         title
                  type     "string"
         description
                  type     "string"
         reference
                  type     "string"
```

6.9 Operation /joindata/{spatialdatasetid}?attributedataseturl

The HTTP GET operation at path **/joindata/{spatialdatasetid}** joins an attribute dataset data with a spatial dataset specified with the **spatialdatasetid** parameter. The attribute dataset data is retrieved from the URL given by the mandatory query parameter **attributedataseturl**.

The operation response contains a list of created outputs and information on the successfulness of the join. The join successfulness information includes following fields:

- numberOfMatchedSpatialDatasetKeys
  - Contains the amount of spatial dataset keys, to which attribute data was successfully joined
- matchedSpatialDatasetKeys
  - Contains a list of spatial dataset keys, to which attribute data was successfully joined
- numberOfUnmatchedSpatialDatasetKeys
  - Contains the amount of spatial dataset keys, to which attribute data could not be joined
- unmatchedSpatialDatasetKeys
  - Contains a list of spatial dataset keys, to which attribute data could not be joined
- numberOfAdditionalAttributeKeys
  - Contains the amount of attribute elements that didn’t have corresponding spatial features available

- AdditionalAttributeKeyList
  - Contains a list of attribute identifiers that didn’t have corresponding spatial features available

6.9.1 Request

<table>
<tr>
<td><b>Req 15</b></td>
<td>/req/core/req-joindata-spatialdatasetid-op

The server SHALL support the HTTP GET operation at the path /joindata/{spatialdatasetid}

<b>Query parameters:</b>

<table>
<tr><th>Name</th><th>Description</th><th>Type and values</th><th>Required</th></tr>
<tr><td>attributedataseturl</td><td>The URL to the attribute dataset data request</td><td>URL type, not empty. The value is a request from Chapter 6.4</td><td>Mandatory</td></tr>
</table>
.</td>
</tr>
</table>

6.9.2 Response

Req 16	/req/core/req-joindata-spatialdatasetid-success A successful execution of the operation SHALL be reported as a response with a HTTP status code 200 The server SHALL return information on the join’s successfulness and links to the created outputs.
	/req/core/req-joindata-spatialdatasetid-failure If the spatial dataset key values are not found on the server, it SHALL be reported as a response with a HTTP status code 404
	/req/core/req-joindata-spatialdatasetid-errpr If there is an error in the server during the processing of the request, it SHALL be reported as a response with a HTTP status code 500

6.9.3 Schema for the /joindata/{spatialdatasetid}?attributedataseturl response

```
JoinDataObject
type    "object"
properties
        inputs
                $ref    "#/definitions/InputsObject"
        outputs
                type    "array"
                items
                        $ref    "#/definitions/OutputObject"
        joinInformation
                $ref    "#/definitions/JoinInformationObject"


InputsObject
type    "object"
properties
        attributeDataset
        type    "string"
        spatialDataset
        type    "string"


OutputObject
type    "object"
properties
        format
                type    "string"
        link
                type    "string"


JoinInformationObject
type    "object"
properties
        numberOfMatchedSpatialDatasetKeys
                type    "integer"
                format  "int32"
        numberOfUnmatchedSpatialDatasetKeys
                type    "integer"
                format  "int32"
        numberOfAdditionalAttributeKeys
                type    "integer"
                format  "int32"
        matchedSpatialDatasetKeys
                type    "array"
                items
                        type    "string"
        unmatchedSpatialDatasetKeys
                type    "array"
                items
                        type    "string"
        additionalAttributeKeysList
                type    "array"
                items
                        type    "string"
```

7. Media Types for any data encoding(s)

A section describing the MIME-types to be used is mandatory for any standard involving data encodings. If no suitable MIME type exists in http://www.iana.org/assignments/media-types/index.html then this section may be used to define a new MIME type for registration with IANA.

Annex A: Conformance Class Abstract Test Suite (Normative)

A.1 Conformance class: AAAA (repeat as necessary)

Annex <insert Annex number>: Revision history

Date	Release	Author	Paragraph modified	Description

Annex <insert annex number>: Bibliography

<A Bibliography, if present, shall appear as the last annex. >