



*International
Virtual
Observatory
Alliance*

Simple Cone Search

Version 1.1

IVOA Working Draft 2019-11-15

Working group

Data Access Layer

This version

<http://www.ivoa.net/documents/ConeSearch/20191115>

Latest version

<http://www.ivoa.net/documents/ConeSearch>

Previous versions

REC 1.03

PR 2007-09-14

PR 2007-06-28

PR 2006-09-08

Author(s)

Marco Molinaro, Ada Nebot, Markus Demleitner, Robert Hanisch,
Raymond Plante, Alex Szalay, Roy Williams

Editor(s)

Marco Molinaro, Ada Nebot

Abstract

This specification defines a simple query protocol, named Simple Cone Search, for retrieving records from a catalog of astronomical sources. The query describes a sky position and an angular distance, defining a cone on the sky. The response returns a list of astronomical sources from the catalog whose positions lie within the cone, formatted as a VOTable. This version aims essentially at aligning this specification with the Data Access Layer (DAL) landscape at the time of writing.

Status of this document

This is an IVOA Working Draft for review by IVOA members and other interested parties. It is a draft document and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use IVOA Working Drafts as reference materials or to cite them as other than “work in progress”.

A list of current IVOA Recommendations and other technical documents can be found at <http://www.ivoa.net/documents/>.

Contents

1	Introduction	3
1.1	Role within the VO Architecture	4
2	Resource	5
2.1	{query} resource	5
2.1.1	RA	6
2.1.2	DEC	6
2.1.3	SR	6
2.1.4	MAXREC	6
2.1.5	RESPONSEFORMAT	7
2.1.6	VERB	7
2.2	Query examples	7
2.3	Availability: VOSI-availability	8
2.4	Capabilities: VOSI-capabilities	8
3	Successful Response	8
3.1	VOTable compliant response	8
4	Error Response	9
4.1	(deprecated) Alternative Error Response	10
5	Resource Registry Extension	11
5.1	Resource record and Capability requirements	11
5.1.1	Capability	12
5.1.2	Interface	12
5.1.3	cs:ConeSearch	12
5.1.4	testQuery and the Query Type	14
A	Sample VOTable Response	15

B Changes from Previous Versions	16
B.1 Changes from REC-1.03	16
B.2 Changes from PR-1.02	17
B.3 Changes from PR-1.01	17
B.4 Changes from PR-1.00	17
B.5 Changes from the original NVO Specification Document . . .	17

Acknowledgments

This document was originally published as a document of the US National Virtual Observatory (NVO) and then transcribed into an IVOA standards document format to become an IVOA recommendation. The changes made in this transcription and in the process of producing the first recommendation are reported in Appendix B.

The Simple Cone Search version 1.03 document has been developed with support from the National Science Foundation’s Information Technology Research Program under Cooperative Agreement AST0122449 with The Johns Hopkins University.

The ConeSearch-1.1 revision has been developed under the ASTERICS project, supported by the European Commission Framework Programme Horizon 2020 Research and Innovation action, grant agreement n. 653477.

Work of the original authors of the Cone Search specification as well as the numerous data providers who have implemented and continue to support this protocol is kindly acknowledged.

Conformance-related definitions

The words “MUST”, “SHALL”, “SHOULD”, “MAY”, “RECOMMENDED”, and “OPTIONAL” (in upper or lower case) used in this document are to be interpreted as described in IETF standard RFC2119 (Bradner, 1997).

The *Virtual Observatory (VO)* is a general term for a collection of federated resources that can be used to conduct astronomical research, education, and outreach. The *International Virtual Observatory Alliance (IVOA)* is a global collaboration of separately funded projects to develop standards and infrastructure that enable VO applications.

1 Introduction

This specification describes how a provider of an astronomical source catalog can publish that catalog to the Virtual Observatory in such a way that a simple cone search can be done. The data remains in the control of the data

check if the document or a link is still available

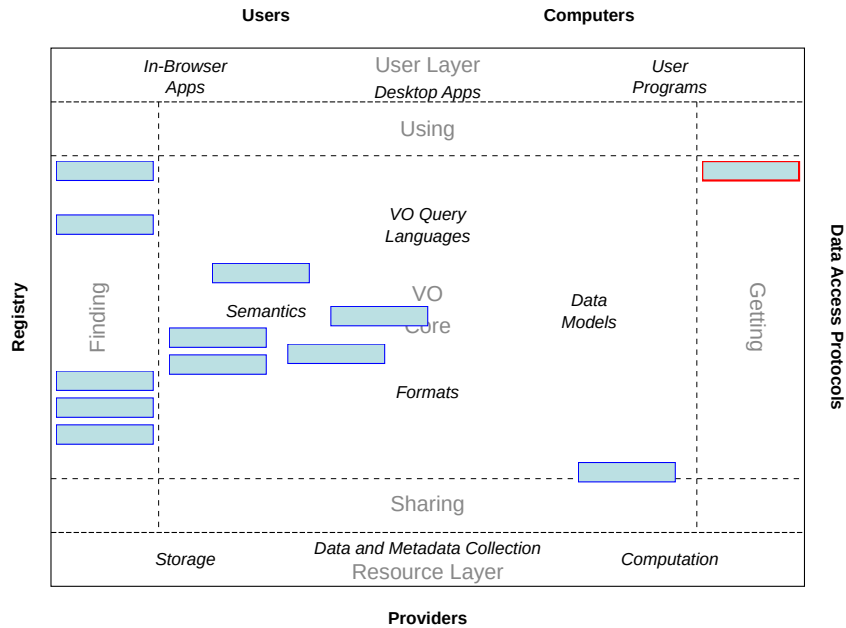


Figure 1: Architecture diagram for Simple Cone Search

provider, served through a web server to the world, but the query profile and response profile are carefully controlled, as described below. It is intended that setting up this web service be simple enough that data providers will not have to spend too much time on it (their funding to support such services is typically small). At the same time, the service implementation and the data it provides can serve as a basis for more sophisticated tools.

This specification does not specify how cone search services are implemented, or how the data are stored or manipulated. The concern of this specification is how the data are exposed to the world through well-defined requests and responses.

This specification assumes that the data provider has already selected a catalog of astronomical sources. This catalog can be presented as a single table; it is expected that the table contains several columns.

1.1 Role within the VO Architecture

Fig. 1 shows the role this document plays within the IVOA architecture (Arviset and Gaudet et al., 2010).

2 Resource

A Simple Cone Search service for accessing catalogue resources is implemented as a synchronous resource, as compliant as possible to the DALI specification (Dowler and Demleitner et al., 2013).

resource type	resource name	required
{query}	service specific	yes
VOSI-capabilities	/capabilities	should
VOSI-availability	(/availability)	should

Table 1: Simple Cone Search resources tree

It **must** have one {query} resource and **should** have a VOSI-capability resource and a VOSI-availability one. The VOSI-capability **must** be a sibling to the {query} one.

2.1 {query} resource

The Simple Cone Search {query} resource **must** meet the following requirements:

- the resource **must** respond to requests submitted using an HTTP GET query string, and **should** respond to HTTP POST query actions (as described in later in this same section);
- the response, by default, **must** be a valid VOTable document, providing a table containing the sources of the catalogue whose positions are within the cone described in the query request (see Sec. 3);
- in case of error, the response **must** follow prescriptions as described in Sec. 4.

The Simple Cone Search service {query} resource is a synchronous web resource, as described by DALI, of the form

`http://<server>/<local_path>?`

To this base URL the query parameters, described hereafter, are attached to build the query request to be submitted through HTTP.

Usage of fixed, custom query parameters, defined by the service provider to build a base URL of the form

`http://<server>/<local_path>?<base_query>&`

(standard behaviour of base URLs in Simple Cone Search-1.03) is allowed but **deprecated** in order to bring cone searches base URLs in the common ReST shape promoted by DALI. Clients **must** anyway treat opaquely these fixed `<base_query>` additions.

In the case of HTTP POST action, the behaviour of Simple Cone Search services having the deprecated base URL format (including custom extra parameters) is not specified. It is however suggested to adopt a mixed GET/POST solution, so the let the fixed `<base_query>` arguments always be submitted in the query part of the HTTP request.

On the opposite, services having plain `{query}` base URLs as mandated by this specification are highly encouraged to support HTTP POST method in order to fulfill DALI compliance.

The set of query constraints a cone search needs to understand **must** include the **RA**, **DEC**, **SR** parameters, **should** include the **MAXREC** and **RESPONSEFORMAT** ones and **may** include the **VERB** one, all of which are to be interpreted with the following stated meaning and the DALI recommendations about parameters.

2.1.1 RA

RA represents a right-ascension in the ICRS coordinate system for the position of the center of the cone to search, given in decimal degrees. It is a single valued parameter and **must** be present in the query.

2.1.2 DEC

DEC represents a declination in the ICRS coordinate system for the position of the center of the cone to search, given in decimal degrees. It is a single value parameter and **must** be present in the query.

2.1.3 SR

SR represents the radius of the cone to search, given in decimal degrees. It is a single valued parameter and **must** be present in the query. If set to zero (SR=0) it should have the same effect as setting MAXREC=0, i.e. a service metadata request as prescribed by DALI¹.

2.1.4 MAXREC

As defined by DALI a cone search **should** accept the **MAXREC** parameter to let the client limit the number of records returned or require a service

¹SR=0 is kept in this version of this specification for back compatibility. It is suggested to prefer the usage of MAXREC to enable metadata discovery

metadata response (see above). Its usage is encouraged and preferred to the SR=0 solution for metadata discovery.

2.1.5 RESPONSEFORMAT

An Simple Cone Search service **must** provide responses in VOTable (Ochsenbein and Williams et al., 2013) format, compliant with respect to what will be detailed in Sec. 3, but **should** also support the DALI **RESPONSEFORMAT** parameter. Allowed media types for VOTable response **should** be the `application/x-votable+xml` or `text/xml` as specified by DALI but `text/xml;content=x-votable` may be considered legal for backward compatibility reasons.

2.1.6 VERB

The query **may** contain the optional single valued parameter, **VERB**, whose value is an integer (either 1, 2, or 3) indicating verbosity which determines how many columns are to be returned in the resulting table. If the service supports the parameter, then when the value is 1, the response should include the bare minimum of columns that the provider considers useful in describing the returned objects (while still remaining compliant with this standard; see section 3 below). When the value is 3, the service should return all of the columns that are available for describing the objects. A value of 2 is intended for requesting a medium number of columns between the minimum and maximum (inclusive) that are considered by the provider to most typically useful to the user. When the VERB parameter is not provided, the server should respond as if VERB=2. If the VERB parameter is not supported by the service, the service should ignore the parameter and should always return the same set of columns for every request.

There may be other parameters in the query, but this document does not specify their meaning or usage. If a query includes an optional parameter, either one specified by this document or not, that is not supported by the service implementation, the service must ignore that parameter.

A query following this syntax represents a request for information on sources located within the specified cone on the sky.

2.2 Query examples

Minimal Simple Cone Search query

```
http://my.cone/search?RA=10.68&DEC=41.26&SR=0.01
```

Service Metadata query

```
http://my.cone/search?cat=A1&RA=0&DEC=0&SR=0&MAXREC=0
```

Limit number of records in response

```
http://my.cone/search?RA=10.68&DEC=41.26&SR=1&MAXREC=100
```

MMo: I don't think I like this idea, it sound like failing silently

Ask for the minimal set of response fields

`http://my.cone/search?RA=10.68&DEC=41.26&SR=0.01&VERB=1`

2.3 Availability: VOSI-availability

A web service with Simple Cone Search capabilities **should** have a VOSI-availability resource as described in DALI. Since VOSI relaxed the availability endpoint, letting it be located elsewhere than being a sibling to the service base URL, support for this interface is encouraged even in the case of services having base URL of the deprecated form.

2.4 Capabilities: VOSI-capabilities

A web service with Simple Cone Search capabilities must have a VOSI-capabilities resource as described in DALI. The standardID for the query capability is reported, among other details and recommendations in Sec. 5.1.1.

Services that present the {query} base URL as a plain ReST resource without additional opaque query parameters are strongly encouraged to provide a capabilities endpoint as a sibling to the {query} resource. The capabilities resource may be unfeasible to maintain for services exposing a base URL with the deprecated format.

3 Successful Response

A successful query **must** result in an HTTP response with status code 200 (OK) and a content that, by default, **must** be in VOTable format (version 1.0 or later), that represents a table of astronomical sources whose positions are within the cone (see Appendix A for an example).

Example needs to be rebuild.

There may also be other sources outside the cone, the service implementor may decide on the exact search criterion used internally to the service to select the sources.

The response format may be other than VOTable when requested using the RESPONSEFORMAT query parameter described in Sec. 2.1.5. Mimetype of the response will vary accordingly to the output format, again as prescribed by DALI and reported in Sec. 2.1.5.

3.1 VOTable compliant response

In the case the service response is serialised in the default, VOTable, format, this response **MUST** comply with the following requirements.

There **must** be a single *RESOURCE* with *type="results"* in the VOTable, and containing a single *TABLE*.

The *TABLE* **must** have *FIELDS* where the following UCD values have been set. There **must** only be one *FIELD* with each of these UCDS:

Do current cone services use this attribute? Update: no, most Simple Cone Search-1.03 services do not.

should we relax on the number of tables? Discuss this with apps.

- **Exactly one *FIELD* must** have *ucd*="ID_MAIN", with an array character type (fixed or variable length), representing an ID string for that record of the table. This identifier **may not** be repeated in the table, and it could be used to retrieve that same record again from that same table.
- **Exactly one *FIELD***, representing the right ascension of the source in the ICRS coordinate system, **must** have *ucd*="POS_EQ_RA_MAIN", and have *datatype* set to "float" or "double".
- **Exactly one *FIELD***, representing the declination of the source in the ICRS coordinate system, **must** have *ucd*="POS_EQ_DEC_MAIN", and have *datatype* set to "float" or "double".

The VOTable may include an expression of the uncertainty of the positions given in the above mentioned fields to be interpreted either as a positional error of the source positions, or an angular size if the sources are resolved. If this uncertainty is not provided, it should be taken to be zero; otherwise, it may be set for all table entries with a *PARAM* in the *RESOURCE* which has a UCD that is set to OBS_ANG-SIZE and has a value which is the angle in decimal degrees. Alternatively, a different value for each row in the table can be given via a *FIELD* in the table having a UCD set to OBS_ANG-SIZE. There may be other *FIELDS* in the table. Their specification **should** include description, datatype, and UCD.

A solution to move to UCD1+ would be nice, but it should probably be postponed to a major revision.

4 Error Response

If the service detects an exceptional condition, it **should** return an error document with an appropriate status code, as specified by DALI, with, possibly, a Content-Type header to tell the client the format of the document.

In the case the error is expressed in VOTable format, recommendation expressed in Section 4.4 of DALI (currently version 1.1) **should** be followed, including the overflow behaviour in the case the MAXREC parameter is in use.

Errors **must** be reported in case any one of the three required parameters (RA, DEC, SR) is missing, or if their values cannot be parsed to floating-point numbers, or if the numbers are out of range (DEC=91.0, for example).

The service **may** also return an error if the search radius (given by the SR parameter) is larger than the one set by the *maxSR* element child of the *capability* one of the service described as a VO resource (see Sec. 5).

The service **may**, as an alternative, report exceptions using the profile expressed by the previous Simple Cone Search recommendation (Williams and Hanisch et al., 2008, v1.03). This alternative for error reporting is to be considered **deprecated**, being the purpose of this specification to fill in the

gap of the Simple Cone Search protocol with the general DAL landscape. The way the alternative works is detailed in Sec. 4.1 here below.

4.1 (deprecated) Alternative Error Response

In the case of error, the service **must** respond with a VOTable that contains a single *PARAM* element or a single *INFO* element with *name*="Error", where the corresponding *value* attribute should contain some explanation of the nature of the error. If an *INFO* element is used, it must appear as a direct child of one of the following elements:

- the root *VOTABLE* element (which is preferred by this document), or
- the *RESOURCE* element

If a *PARAM* element is used, it must appear as a direct child of the *RESOURCE* element. Please note that, apart from being deprecated here, the use of the *PARAM* element to convey error response was already discouraged in the previous version of this specification.

Example Error Responses

Error INFO as child of VOTABLE (preferred)

```
<?xml version="1.0"?>
<!DOCTYPE VOTABLE SYSTEM "http://us-vo.org/xml/VOTable.dtd">
<VOTABLE version="1.0">
  <DESCRIPTION>MAST Simple Cone Search Service</DESCRIPTION>
  <INFO ID="Error" name="Error" value="Error in input RA value: as3f"/>
</VOTABLE>
```

Error PARAM as child of RESOURCE (allowed)

```
<?xml version="1.0"?>
<!DOCTYPE VOTABLE SYSTEM "http://us-vo.org/xml/VOTable.dtd">
<VOTABLE version="1.0">
  <DESCRIPTION>
    HEASARC Browse data service
    Please send inquiries to mailto:request@athena.gsfc.nasa.gov
  </DESCRIPTION>
  <RESOURCE ID="error_resource">
    <PARAM ID="Error" name="Error" datatype="char" arraysize="*"
      value="Invalid data type: text/html"/>
  </RESOURCE>
</VOTABLE>
```

Queries targeting no records **should not** generate an error response, but an empty metadata response.

Removed reference to the DEFINITION element solution, definitely out of VOTable since long.

Does it make sense to keep this sentence, shortened from the previous version? MMo guess: no

Note

The content of **Section 5** is taken from REC-SimpleDALRegExt-1.1, sections 1, 2 and 3, specifically section 3.1 that is dedicated to the Simple Cone Search case.

5 Resource Registry Extension

To adequately describe that a service supports the Simple Cone Search protocol, it is necessary to define Simple Cone Search specific capability metadata. This is needed both to allow discovery of cone search services within VO registered resources (through the Registry Interfaces standard, [Benson and Plante et al. \(2009\)](#)), deploying VOResource documents, [Plante and Benson et al. \(2008\)](#)) and to generally describe service behaviour to help applications consume it properly, given compliance to the protocol.

This section specifies these metadata for cone search resources and is intended to be applicable wherever VOResource records are used, in particular for standard encoding of resource descriptions within IVOA registries and for encoding capability metadata available through VOSI ([Grid and Web Services Working Group, 2017](#)) interfaces.

Apart from the above standards referenced here above, this registry extension depends on the VODataService ([Plante and Stébé et al., 2010](#)) standard.

5.1 Resource record and Capability requirements

To be recognized as a Simple Cone Search, the service resource must be described as a resource of type *vr:Service* (defined in the VOResource schema) or one of its legal sub-types. The resource type is set by setting the *xsi:type* attribute on the element representing the root of the VOResource record to the namespace-qualified resource type name. Simple Cone Search **should** be of the resource type *vs:CatalogService* (defined in the VODataService extension schema)².

Since the *vs:CatalogService* resource type allows it, record authors are encouraged to include a full description of the columns in the table returned in query response (assuming full verbosity), as well as to provide sky coverage information.

²*vr:* and *vs:* are the canonical prefixes for the namespaces associated with VOResource and VODataService XML schemata

5.1.1 Capability

The VOResource record **must** include a *capability* element that **must** have a *standardID* attribute set to

```
ivo://ivoa.net/std/conesearch#query-1.1
```

to unambiguously identify the resource as a Simple Cone Search compliant to Simple Cone Search-1.1 version. The *capability* **should** also have *xsi:type="cs:ConeSearch"* to specialize the *vr:Capability* to be of the specific sub-type supporting the cone search protocols. The *cs:* here refers to the canonical prefix for the namespace associated with the Simple Cone Search extension schema, that is

```
http://www.ivoa.net/xml/ConeSearch/v1.0 ,
```

the same as for Simple Cone Search-1.03, as explained in the XML Schema Versioning (Harrison and Demleitner et al., 2016) document; distinction among the two schemata version being delivered by the *version* attribute in the schema root element.

The *cs:ConeSearch* type is described in Sec. 5.1.3.

5.1.2 Interface

The *capability* element described in Sec. 5.1.1 **must** include a child *interface* element. The *interface* element *xsi:type* attribute **must** be set to *vs:ParamHTTP*, and its *role* attribute **must** be set to "std". A *accessURL* element within that *interface* **must** be set to the *<base_url>* of the service (see 2.1).

It is not necessary to provide the *use* attribute to the *accessURL* element (as its value can be assumed); however, when it is provided, it must be set to "base". Similarly, it is not necessary to provide the *interface* element with *queryType* or *resultType* elements; however, when provided, their values should be "GET" and "application/x-votable+xml", respectively. The *vs:ParamHTTP* allows one to describe input parameters supported by the service; description authors are encouraged to list the optional parameters and any custom parameters supported by the instance of the service.

5.1.3 cs:ConeSearch

The *cs:ConeSearch* type is a *vr:Capability* sub-type that should be used to describe a service's support for the Simple Cone Search protocol; it is defined as follows:

cs:ConeSearch Type Schema Documentation

The capabilities of a Cone Search implementation.

cs:ConeSearch Type Schema Definition

```
<xs:complexType name="ConeSearch" >
  <xs:complexContent >
    <xs:extension base="vr:Capability" >
      <xs:sequence >
        <xs:element name="maxSR" type="xs:float" minOccurs="0"
          maxOccurs="1" />
        <xs:element name="maxRecords" type="xs:positiveInteger"
          minOccurs="0" maxOccurs="1" />
        <xs:element name="verbosity" type="xs:boolean" />
        <xs:element name="testQuery" type="cs:Query" minOccurs="0"
          maxOccurs="1" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

cs:ConeSearch Extension Metadata Elements

Element *maxSR*

Type floating-point number: *xs:float*

Meaning The largest search radius, in degrees, that will be accepted by the service without returning an error condition. Not providing this element or specifying a value of 180 indicates that there is no restriction.

Occurrence optional

Comment Not providing a value is the preferred way to indicate that there is no restriction.

Element *maxRecords*

Type *xs:positiveInteger*

Meaning The largest number of records that the service will return. Not providing this value means that there is no effective limit.

Occurrence optional

Comment This does not refer to the total number of records in the catalog but rather maximum number of records the service is capable of returning. A limit that is greater than the number of records available in the archive is equivalent to their being no effective limit. (See RM, Hanisch 2007.)

Element *verbosity*

Type boolean (true/false): *xs:boolean*

Meaning True if the service supports the VERB keyword; false, otherwise.

Occurrence required

Element *testQuery*

Type composite: *cs:Query*

Meaning A query that will result in at least one matched record that can be used to test the service.

Occurrence optional

The custom metadata that the *cs:ConeSearch* type provides is given above. Other general metadata useful to describe the Simple Cone Search specification are directly part of the core VOResource schema.

5.1.4 testQuery and the Query Type

The *testQuery* element is intended to help other VO components (e.g. registries, validation services, services that monitor the VO's operational health, but typically not end users) test that the service is up and operating correctly. It provides a set of legal input parameters that should return a legal response that includes at least one matched record. Since this query is intended for testing purposes, the size of the result set should be small.

The *cs:Query* type captures the different components of the query into separate elements, as defined below:

cs:Query Type Schema Documentation

A query to be sent to the service

cs:Query Type Schema Definition

```
<xs:complexType name="Query" >
  <xs:sequence >
    <xs:element name="ra" type="xs:double" />
    <xs:element name="dec" type="xs:double" />
    <xs:element name="sr" type="xs:double" />
    <xs:element name="verb" type="xs:positiveInteger" minOccurs="0" />
    <xs:element name="catalog" type="xs:string" minOccurs="0" />
    <xs:element name="extras" type="xs:string" minOccurs="0" />
  </xs:sequence>
</xs:complexType>
```

cs:Query Metadata Elements

Element *ra*

Type floating-point number: *xs:double*

Meaning the right ascension of the search cone's center in decimal degrees.

Occurrence required

Element *dec*

Type floating-point number: *xs:double*

Meaning the declination of the search cone's center in decimal degrees.

Occurrence required

Element *sr*

Type floating-point number: *xs:double*

Meaning the radius of the search cone in decimal degrees.

Occurrence required

Element *verb*

Type *xs:positiveInteger*

Meaning the verbosity level to use where 1 means the bare minimum set of columns and 3 means the full set of available columns.

Occurrence optional

Element *catalog*

Type string: *xs:string*

Meaning the catalog to query.

Occurrence optional

Comment When the service can access more than one catalog, this input parameter, if available, is used to indicate which service to access.

Element *extras*

Type string: *xs:string*

Meaning any extra (non-standard) parameters that must be provided (apart from what is part of base URL given by the accessURL element).

Occurrence optional

Comment this value should be in the form of name=value pairs delimited with ampersands (&).

attach xsd?

A Sample VOTable Response

Needs to be changed

This is a sample of a legal response to a Cone Search query.

```
<?xml version="1.0"?>
<!DOCTYPE VOTABLE SYSTEM "http://us-vo.org/xml/VOTable.dtd">
<VOTABLE version="1.0">
  <DEFINITIONS>
    <COOSYS system="eq_FK5" equinox="2000" />
  </DEFINITIONS>
  <RESOURCE ID="T9001">
    <DESCRIPTION>
      HEASARC Browse data service
      Please send inquiries to mailto:request@athena.gsfc.nasa.gov
    </DESCRIPTION>
    <PARAM ID="default_search_radius" ucd="OBS_ANG-SIZE" datatype="double"
      value="0.05166666666666667" />
    <TABLE ID="heasarc_first_9001">
      <DESCRIPTION> Faint Images of the Radio Sky at Twenty cm Source Catalog (FIRST) </DESCRIPTION>
      <FIELD name="unique_id" datatype="char" arraysize="*" ucd="ID_MAIN">
        <DESCRIPTION> Integer key </DESCRIPTION>
      </FIELD>
      <FIELD name="name" datatype="char" arraysize="*" >
        <DESCRIPTION> FIRST Source Designation </DESCRIPTION>
      </FIELD>
      <FIELD name="ra" datatype="double" unit="degree" ucd="POS_EQ_RA_MAIN">
```

```

    <DESCRIPTION> Right Ascension </DESCRIPTION>
</FIELD>
<FIELD name="dec" datatype="double" unit="degree" ucd="POS_EQ_DEC_MAIN">
  <DESCRIPTION> Declination </DESCRIPTION>
</FIELD>
<FIELD name="flux_20_cm" datatype="double" unit="mJy" >
  <DESCRIPTION> Peak 1.4GHz Flux Density (mJy) </DESCRIPTION>
</FIELD>
<FIELD name="flux_20_cm_error" datatype="double" unit="mJy" >
  <DESCRIPTION> Estimated rms in at Source (mJy) </DESCRIPTION>
</FIELD>
<FIELD name="int_flux_20_cm" datatype="double" unit="mJy" >
  <DESCRIPTION> Integrated 1.4GHz Flux Density (mJy) </DESCRIPTION>
</FIELD>
<DATA>
  <TABLEDATA>
    <TR>
      <TD>384559</TD><TD>FIRST J120002.6+595708</TD>
      <TD>180.0110042</TD><TD>59.9523889</TD>
      <TD> 1.11</TD><TD> 0.139</TD><TD> 1.14</TD>
    </TR>
    <TR>
      <TD>385094</TD><TD>FIRST J120025.3+600103</TD>
      <TD>180.1057250</TD><TD>60.0175556</TD>
      <TD> 2.89</TD><TD> 0.142</TD><TD> 2.56</TD>
    </TR>
    <TR>
      <TD>384928</TD><TD>FIRST J120018.1+600236</TD>
      <TD>180.0755500</TD><TD>60.0434750</TD>
      <TD> 19.38</TD><TD> 0.145</TD><TD> 19.23</TD>
    </TR>
    <TR>
      <TD>384490</TD><TD>FIRST J115959.4+600403</TD>
      <TD>179.9978875</TD><TD>60.0677083</TD>
      <TD> 1.01</TD><TD> 0.147</TD><TD> 1.20</TD>
    </TR>
  </TABLEDATA>
</DATA>
</TABLE>
</RESOURCE>
</VOTABLE>

```

B Changes from Previous Versions

B.1 Changes from REC-1.03

- moving base URL to a plain `http://server/path?` format
- changed error response to comply with DALI
- changed resource metadata importing directly from SimpleDALRegExt
- relaxed RA and Dec FIELDS in response to allow float or double datatype

- no more exactly one RESOURCE in response, now stating exactly one of type="results"
- removed fixed version (1.0 or 1.1) for VOTable default response
- Added DALI MAXREC and RESPONSEFORMAT
- Added POST as optional HTTP query method
- Addition and changes on authors/editors.
- Plain porting of the HTML document into ivoatex template, including change history, then modified it and reshaped.

B.2 Changes from PR-1.02

- converted to Recommendation

B.3 Changes from PR-1.01

- eliminated the choice of encoding an ERROR response in a PARAM that is a direct child of VOTABLE as this is not legal in the VOTable schema.
- Allowed the use of the INFO element for error messages.
- In examples, made sure PARAM elements have datatype attributes
- Corrected wording to be definitive that positions are given in the ICRS coordinate system.
- Other typos.

B.4 Changes from PR-1.00

- Various typos.
- Clarified description of VERB parameter:
 - Clarified what is meant by optional for client and server.
 - Clarified the meaning of the values.
- Explicitly mention the 3 legal locations for ERROR messages.
- Refer to string types as character arrays, as per the VOTable std.
- Prefers text/xml;content=x-votable over text/xml;votable.
- Added examples of error message, legal response in appendix.

B.5 Changes from the original NVO Specification Document

- References to the original HTML document have been replaced with references to this IVOA specification.
- Replaced references to "curator" with "data provider" or similar wording.
- Replaced references to the NVO with references either to the IVOA or this specification, as appropriate.

- Ambiguous language like "perhaps" has been replaced with more definitive wording where appropriate. Use of the word "will" is replaced with "must" and "can" with "may", in accordance with the definitions set in the preface.
- Grammatical and spelling corrections have been made.
- The content is organized into sections typical of an IVOA spec.
- Description of the URL syntax is sharper, borrowing language from the SIA specification [SIA]. This includes:
 - More explicitly specifying the form of the URL.
 - Sharpening the definition of the input parameters.
 - Indicating that parameter order is not significant.
 - Stating explicitly that unsupported optional arguments should be ignored.
 - Adding examples.
 - Re-ordering information for improved flow.
- The version of VOTable supported is explicitly stated.
- Whereas the NVO version describes the parameter with `ucd="OBS_ANG-SIZE"` as "an expression of the opening angle of the cones", this version describes it specifically as "an expression of the uncertainty of the positions".
- A note has been added to recognize the ambiguity in the location of the `ERROR` parameter.
- The general description of the resource profile has been altered to allow rendering of the metadata to change according to the standards and conventions of the IVOA.
- An editor's note has been added that makes reference to the RM document [RM].
- A requirement that **MaxSR** be given in decimal degrees has been added.
- For the **BaseURL** resource profile metadata, the example has been replaced with a reference to the BaseURL syntax description.
- An appendix has been added to describe the current practice for registering Cone Search services.

References

- Arviset, C., Gaudet, S. and the IVOA Technical Coordination Group (2010), 'IVOA architecture', IVOA Note.
<http://www.ivoa.net/documents/Notes/IVOOArchitecture>
- Benson, K., Plante, R., Auden, E., Graham, M., Greene, G., Hill, M., Linde, T., Morris, D., O'Mullane, W., Rixon, G., Stébé, A. and Andrews, K.

- (2009), ‘IVOA registry interfaces version 1.0’, IVOA Recommendation.
<http://www.ivoa.net/documents/RegistryInterface/>
- Bradner, S. (1997), ‘Key words for use in RFCs to indicate requirement levels’, RFC 2119.
<http://www.ietf.org/rfc/rfc2119.txt>
- Dowler, P., Demleitner, M., Taylor, M. and Tody, D. (2013), ‘Data access layer interface, version 1.0’, IVOA Recommendation.
<http://www.ivoa.net/documents/DALI/20131129/>
- Grid and Web Services Working Group (2017), ‘IVOA support interfaces version 1.1’.
<http://www.ivoa.net/documents/VOSI/20170524/index.html>
- Harrison, P., Demleitner, M., Major, B. and Dowler, P. (2016), ‘XML schema versioning policies’, IVOA Proposed Endorsed Note.
<http://ivoa.net/documents/Notes/XMLVers>
- Ochsenbein, F., Williams, R., Davenhall, C., Demleitner, M., Durand, D., Fernique, P., Giarretta, D., Hanisch, R., McGlynn, T., Szalay, A., Taylor, M. and Wicenec, A. (2013), ‘VOTable format definition, version 1.3’, IVOA Recommendation.
<http://www.ivoa.net/documents/VOTable/>
- Plante, R., Benson, K., Graham, M., Greene, G., Harrison, P., Lemson, G., Linde, T., Rixon, G. and Stébé, A. (2008), ‘VOResource: an XML encoding schema for resource metadata version 1.03’, IVOA Recommendation.
<http://www.ivoa.net/documents/REC/ReR/VOResource-20080222.html>
- Plante, R., Stébé, A., Benson, K., Dowler, P., Graham, M., Greene, G., Harrison, P., Lemson, G., Linde, T. and Rixon, G. (2010), ‘VODataService: a VOResource schema extension for describing collections and services version 1.1’, IVOA Recommendation.
<http://www.ivoa.net/documents/VODataService/>
- Williams, R., Hanisch, R., Szalay, A. and Plante, R. (2008), ‘Simple cone search - version 1.03’, IVOA Recommendation.
<http://www.ivoa.net/documents/latest/ConeSearch.html>