



IVOA Query Interface Version 0.2

**IVOA Working Draft
2003-08-07**

This version:

Latest version:

Previous versions:
none

Editors:
Masatoshi Ohishi

Authors:
William O'Mullane, Vivek Haridas, Naoki Yasuda

Abstract

This document describes the minimum required interface to participate in the IVOA as a queryable node.

Status of this document

This is a Working Draft. There are no prior released versions 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](http://www.ivoa.net/docs/) can be found at <http://www.ivoa.net/docs/>.

Acknowledgments

This work is based on discussions at the Cambridge IVOA meeting.

Contents

Abstract.....	1
Status of this document.....	1
Acknowledgments.....	2
Contents.....	2
1 Introduction	2
2 IVOA Query Interface.....	2
2.1 Error Handling	2
2.2 Standard VO Interface.....	3
2.3 Astronomical Data Query Language(ADQL)	3
2.4 Query Interface.....	3
3 Changes from previous versions.....	4

1 Introduction

At the Cambridge IVOA meeting there was general agreement to the ADQL concept. This implied an SQL like language but passed as an XML document. VOQL (Virtual Observatory Query Language) was seen as a higher layer built on ADQL and web services.

We have now demonstrated that a single WSDL (Web Services Definition Language) file may be implemented in at least Java and C#, i.e. on Linux as well as windows. Also we have demonstrated that a single client may interact with both implementations of the service.

In this document we would like to propose the astronomical data query interface for the virtual observatory. It is presented here as a set of Services which we believe any Virtual Observatory Service should implement and the specific services a Query Interface should implement.

Finally we include a link the WSDL for the proposed services. It is not included as an appendix here as it comes to some 25 pages and is really meant for use with SOAP tools rather than for human consumption.

2 IVOA Query Interface

2.1 Error Handling

All errors in the services should be returned as SOAP exceptions. There is no perceived need at this time to have special Exceptions although this may become useful in the future. In the initial version then all Exceptions may simply be a useful message in a generic Exception.

2.2 Standard VO Interface

For WebServices the service should be the single authoritative source for metadata. Currently services provide WSDL in a standard manner. WSDL alone is insufficient for the purposes of the VO. We need something more like the VOResource implementation of the RSM (Resource & Metadata) document. In a web service we would return the object type produced from the XSD for the RSM. The definition of Voresource must be included in the WSDL for the service. This does not preclude returning an extension of the VOResource. This may be seen more clearly by looking at the dummy service <http://skyservice.pha.jhu.edu/devel/adqlarchive/archiveservice.asmx>.

- QI-1** All VO WebServices shall implement the “MetaData” service. This service shall return a valid VOResource describing the metadata of this service.

Note: The metadata for the tables /data in the system are accessed as described below.

2.3 Astronomical Data Query Language(ADQL)

ADQL is passed as an XML document to the Query Interface. ADQL is based on a subset of SQL plus Circle(Cone Search). The only SQL command allowed in ADQL is a “select”. The full XSD for ADQL and services for translation of SQL to ADQL and back may be found at

<http://skyservice.pha.jhu.edu/develop/vo/adql/>

- QI-2** ADQL shall be in the form of an XML document.
- QI-3** All table names in ADQL shall be followed by an alias.
- QI-4** ADQL shall support Column aliasing using the AS keyword.
- QI-5** ADQL shall support the CIRCLE operator to define an area. This shall have the format
CIRCLE CoordSys CoordType n n [n] radius
- QI-6** Mathematical functions shall be allowed in ADQL as follows:

Trigonometric functions: acos, asin, atan, cos, cot, sin, tan

Math functions: abs, ceil, degree, exp, floor, ln, log, mod, pi, pow, radians, sqrt, trunc

2.4 Query Interface

The query interface requires several methods to work. Again several of these methods are for Metadata – in this case we need to know about Tables and Columns that may be included in a query. To facilitate interoperation we will also need to be able to get UCDs for each column. Another service will need to take an ADQL document and return a VOTable of data. The current incarnation of ADQL syntax supports only queries which would return data i.e. there is no provision for Data Manipulation Language which would have no data to return.

- QI-7** A queryable VO service shall implement the “Tables” service, which returns a list of all tables that may be used in ADQL passed to this endpoint.
- QI-8** A queryable VO service shall implement the “Columns” service, which returns information about the columns of a given table name. This shall return an array of MetaColumns this structure shall contain the Column Name, Unit, and UCD.

- QI-9** A queryable VO service shall implement the “PerformQuery” service, which takes an ADQL document as a parameter and returns a VOTable, which is the result of processing the query.

The Latest WSDL file can be located at

<http://skyservice.pha.jhu.edu/devel/adqlArchive/ArchiveService.asmx?wsdl>

Sample applications and tutorials for development and deployment of ADQL services is available at <http://skyservice.pha.jhu.edu/develop/vo/adql/>

3 Changes from previous versions

- Added section on ADQL.
- Changed Columns to use MetaColumn.