



ADQL 2.1  
Progress report  
October 2017

D.Morris  
Institute for Astronomy,  
Edinburgh University  
October 2017



IVOA Interop 2017/10/27



Removed REGION

Removed 'Geometry in the SELECT clause'

“Geometry values ... may be listed in the SELECT clause, in which case they must be converted into a text form.”

```
SELECT circle(ra, dec, 1.0) FROM ...
```

```
"CIRCLE(25.4, -20.0, 1.0)"
```

```
"CIRCLE(19.1, -15.5, 1.0)"
```

```
"CIRCLE(12.0, -14.8, 1.0)"
```





Coordsys param deprecated in all of the geom functions.

`box('ICRS', point, point)`      ➡      `box(point, point)`  
`circle('ICRS', point, radius)`      ➡      `circle(point, radius)`

Need to update the examples in all our other documents.

ObsCore:

```
SELECT * FROM ivoa.ObSCORE WHERE  
CONTAINS(POINT('ICRS',16.0,40.0),s_region)=1
```

TAP:

```
POINT('ICRS', t.ra, t.dec)
```





Expanded the data types from single table into separate sub-sections.

VOTable			ADQL
datatype	arraysize	xtype	type
boolean	1	-	BOOLEAN
short	1	-	SMALLINT
int	1	-	INTEGER
long	1	-	BIGINT
float	1	-	REAL
double	1	-	DOUBLE
(numeric)	> 1	-	implementation defined
char	1	-	CHAR(1)
char	n	-	CHAR(n)
char	n*	-	VARCHAR(n)
unsignedByte	n	-	BINARY(n)
unsignedByte	n*	-	VARBINARY(n)
unsignedByte	n, *, n*	adql:BLOB	BLOB
char	n, *, n*	adql:CLOB	CLOB
char	n, *, n*	adql:TIMESTAMP	TIMESTAMP
char	n, *, n*	adql:POINT	POINT
char	n, *, n*	adql:REGION	REGION

- Logical types
  - Boolean
- Numeric types
  - Numeric primitives
  - INTERVAL
- Date and time
  - TIMESTAMP
- Character types
  - Character primitives
  - CLOB
- Binary types
  - BLOB
- Geometric types
  - POINT
  - CIRCLE
  - POLYGON

Updated to match  
the changes in  
DALI.





## TIMESTAMP

Changed from character based string into a date time value.

*"The ADQL string concatenation operator || should be applicable to all columns resulting from VOTable char-typed columns."*

As a string:

`TIMESTAMP('2015-01-01')` **!=** `TIMESTAMP('2015-01-01T00:00:00')`

As a time:

`TIMESTAMP('2015-01-01')` **==** `TIMESTAMP('2015-01-01T00:00:00')`





## INTERVAL

Defined in DALI, not defined in ADQL.

Do we want an INTERVAL in ADQL ?

Do we want to be able to do this :

```
WHERE ra BETWEEN <interval>
```





## TIMEINTERVAL

DALI defines INTERVAL serialization as array an array of <numeric>:

```
<FIELD datatype="int", arraysize="2", xtype="interval">
```

DALI defines TIMESTAMP serialization as an array of char:

```
<FIELD datatype="char", arraysize="*", xtype="timestamp">
```

How do we represent an interval of time ?





## TIMEINTERVAL

Propose a new xtype of `timeinterval`, serialized as an array of `TIMESTAMP`.

```
<FIELD datatype="char", arraysize="2x*", xtype="timeinterval">
```

```
"1970-01-01T00:00:00.000Z" "2017-08-16T17:12:54.621Z"
```

Enabling us to do this :

```
WHERE t_min BETWEEN <time-interval>
```







Consistency between documents is an issue  
First impressions matter

New joiners are presented with a huge pile of documents.  
Things like data types in examples are inconsistent.  
Adding errata makes it worse.

I spent several days  
tracking the links  
between data types  
in our specifications.

New joiner would have  
no chance.

Propose a small team to review data types and examples  
across our documents.

e.g. VODataService, VOTable, DALI, TAP and ADQL

Only fixing data types and examples.

Creating new minor versions, not errata.





Consistency between documents is an issue  
First impressions matter

VODataService  
TAPType

Duplicates the  
DAL/ADQL  
types

BOOLEAN  
SMALLINT  
INTEGER  
BIGINT  
REAL  
DOUBLE  
TIMESTAMP  
CHAR  
VARCHAR  
BINARY  
VARBINARY  
POINT  
REGION  
CLOB  
BLOB

Not just changes to the  
VODataService specification.

TAPType appears in  
examples in several  
specifications.

VODataService  
extendedType

looks like it is equivalent to

VOTable  
xtype

.. but we never actually say so





## Plans:

New WD available after this meeting.

No new changes planned, send to PR asap.

Start working on PEG grammar for 3.x

