

# Registry Interfaces 1.0: A Service Consumer's Experience

Mark Taylor (Bristol)

IVOA Interop,  
Victoria  
20 May 2010

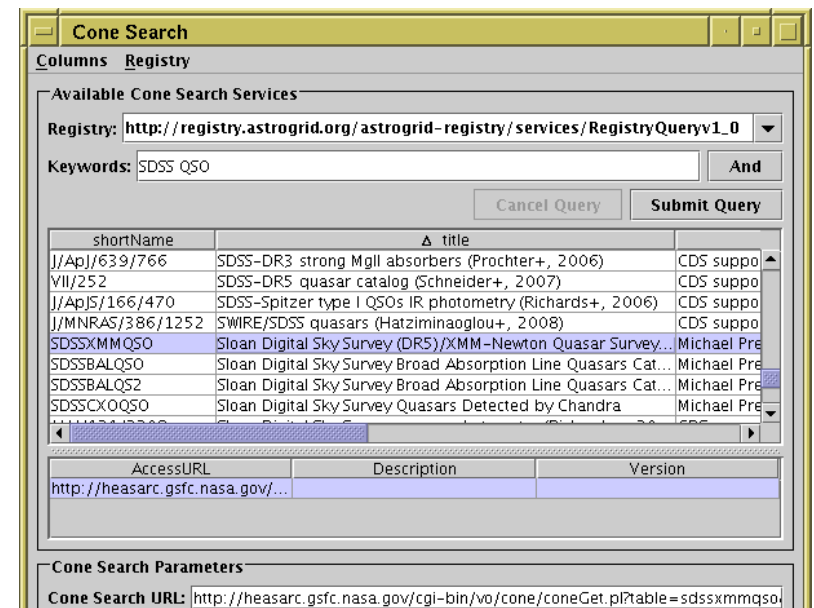
`$Id: reg.tex,v 1.4 2010/05/18 05:52:19 mbt Exp $`

# Requirements

## TOPCAT registry requirements:

- Allow user choice of registry endpoint URL
  - ▷ default, from list, or text entry
- Get list of available DAL services
  - ▷ restricted by type (Cone, SIA, SSA)
  - ▷ optionally restricted by user-supplied keyword(s)
- Display *limited* information about each service

Probably other tools have similar needs



The screenshot shows a web interface titled "Cone Search". It has a search form with a "Registry" dropdown set to "http://registry.astrogrid.org/astrogrid-registry/services/RegistryQueryv1\_0" and "Keywords" set to "SDSS QSO". There are "Cancel Query" and "Submit Query" buttons. Below the form is a table of available services:

shortName	title	
J/ApJ/639/766	SDSS-DR3 strong MgII absorbers (Prochter+, 2006)	CDS suppo
VII/252	SDSS-DR5 quasar catalog (Schneider+, 2007)	CDS suppo
J/ApJ/166/470	SDSS-Spitzer type I QSOs IR photometry (Richards+, 2006)	CDS suppo
J/MNRAS/386/1252	SWIRE/SDSS quasars (Hatziminaoglou+, 2008)	CDS suppo
SDSSXMMQSO	Sloan Digital Sky Survey (DR5)/XMM-Newton Quasar Survey...	Michael Pre
SDSSBALQSO	Sloan Digital Sky Survey Broad Absorption Line Quasars Cat...	Michael Pre
SDSSBALQSO2	Sloan Digital Sky Survey Broad Absorption Line Quasars Cat...	Michael Pre
SDSSCXOQSO	Sloan Digital Sky Survey Quasars Detected by Chandra	Michael Pre

Below the table is a section for "Cone Search Parameters" with a "Cone Search URL" field containing "http://heasarc.gsfc.nasa.gov/cgi-bin/v0/cone/coneGet.pl?table=sdssxmmqso".

# Choice of Service

- Use Registry Interfaces 1.0 standard
  - Standard interface allows use of any compliant VO searchable registry
- Available search SOAP operations:
  - **Search** — free-form ADQL query
  - **KeywordSearch** — search on keywords
    - 😊 Convenient (and efficient?) way to search by keyword
    - 😞 No way to restrict query further (e.g. by service type)
  - **XQuerySearch** — free-form XQuery
    - 😊 Detailed control over which resource metadata items are returned
    - 😞 Optional in RI 1.0
- Only **Search** is suitable

# Registry Search Query

## Search for Cone Search services related to 2MASS

- ADQL/S:

```
Search WHERE ( capability/@standardID = 'ivo://ivoa.net/std/ConeSearch' ) AND
( ( identifier LIKE '%2MASS%' OR
  content/description LIKE '%2MASS%' OR
  title LIKE '%2MASS%' OR
  content/subject LIKE '%2MASS%' OR
  content/type LIKE '%2MASS%' ) )
```

- Issues:

- ▶ How best to search for particular DAL service types?

- `capability/@standardID = 'ivo://ivoa.net/std/ConeSearch' ?`
- `capability/@xsi:type = 'cs:ConeSearch' ?`
- `capability/@xsi:type LIKE ':%ConeSearch' ?`

- ▶ Case sensitivity!

- It's undefined whether LIKE matching is case sensitive; some are (Euro-VO), some are not (NVO, AG)

- ▶ Which fields to match against?

- Reasonably easy to make sensible choices

# Registry Search Query

## Search for Cone Search services related to 2MASS

- ADQL/X:

```
<Where xmlns:ad="http://www.ivoa.net/xml/ADQL/v1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <ad:Condition xsi:type="ad:intersectionSearchType">
    <ad:Condition xsi:type="ad:closedSearchType">
      <ad:Condition xsi:type="ad:comparisonPredType" Comparison="=">
        <ad:Arg xsi:type="ad:columnReferenceType" Table="" xpathName="capability/@standardID" name="@standardID">
          <ad:Arg xsi:type="ad:atomType">
            <ad:Literal xsi:type="ad:stringType" Value="ivo://ivoa.net/std/ConeSearch"/>
          </ad:Arg>
        </ad:Condition>
      </ad:Condition>
    </ad:Condition>
    <ad:Condition xsi:type="ad:closedSearchType">
      <ad:Condition xsi:type="ad:closedSearchType">
        <ad:Condition xsi:type="ad:unionSearchType">
          <ad:Condition xsi:type="ad:unionSearchType">
            <ad:Condition xsi:type="ad:unionSearchType">
              <ad:Condition xsi:type="ad:unionSearchType">
                <ad:Condition xsi:type="ad:likePredType">
                  <ad:Arg xsi:type="ad:columnReferenceType" Table="" xpathName="identifier" name="identifier"/>
                  <ad:Pattern xsi:type="ad:atomType">
                    <ad:Literal xsi:type="ad:stringType" Value="%2MASS%"/>
                  </ad:Pattern>
                </ad:Condition>
              </ad:Condition>
            <ad:Condition xsi:type="ad:likePredType">
              <ad:Arg xsi:type="ad:columnReferenceType" Table="" xpathName="content/description" name="description"/>
              <ad:Pattern xsi:type="ad:atomType">
                <ad:Literal xsi:type="ad:stringType" Value="%2MASS%"/>
              </ad:Pattern>
            </ad:Condition>
          </ad:Condition>
        </ad:Condition>
      <ad:Condition xsi:type="ad:likePredType">
        <ad:Arg xsi:type="ad:columnReferenceType" Table="" xpathName="title" name="title"/>
        <ad:Pattern xsi:type="ad:atomType">
          <ad:Literal xsi:type="ad:stringType" Value="%2MASS%"/>
        </ad:Pattern>
      </ad:Condition>
    </ad:Condition>
  </ad:Condition>
  ....

```

# Registry Access Implementation

- Used Ray Plante's [IVOARegistry](#) library
  - Mostly good
    - ▷ takes care of ADQL/S → ADQL/X — *phew*
    - ▷ one or two small bugs, easy to fix or work around
  - Builds DOM for registry response
    - ▷ Memory hungry for large search responses
    - ▷ Can break up request into several sections — but how big?
    - ▷ Some single resource records are very large (e.g. UKIDSS at WFAU 1Mb XML → several Mb DOM per resource)
- Ended up writing custom lightweight registry SOAP client
  - Uses IVOARegistry lib for ADQL/S→ADQL/X translation
  - Uses small DIY SOAP client (Axis not required - 500 lines is better than 200k)
  - Extracts required resource info from response stream using SAX
  - Fast, very low memory footprint
  - Could be packaged for 3<sup>rd</sup> party use — any takers?

# Optional Extra: Bootstrap Registry List

## What registries are available for searching?

- Would be nice to get/update this list from the registry
  - ▶ RofR no good — that's for harvestable registries
  - ▶ Can query a registry for other searchable registries:

```
Search WHERE capability/@standardID = 'ivo://ivoa.net/std/Registry' AND
        capability/@xsi:type LIKE ':%:Search' AND
        full LIKE 'true'
```

    - NVO reg: no records returned
    - Euro-VO reg: SOAP fault
    - AG reg: returns 4 new registries 😊; none of them work ☹
  - ▶ so removed "Update Registry List" button from the GUI
- Gave up and hardwired AG, NVO, Euro-VO registries as default options instead

# Conclusions

- Based on my experiences:
  - It works — but took some experimentation to get right
  - Initially some services didn't comply with standard — much better now
  - Not very obvious how to write common queries (*"gimme all SSA services"*)
    - ▷ but reg people have been helpful
    - ▷ maybe provide a Howto/Examples list somewhere?
  - Case-sensitivity in **Search** is a serious problem where encountered
    - ▷ Only way to locate all Simbad-related Cone Search services is to try "simbad", "Simbad", "SIMBAD", . . . .
- I could make lightweight registry access library available if demand
- Are other people consuming these services?