1. Async “AccessData”

Markus Demleitner
msdemlei@ari.uni-heidelberg.de

This is a brief report on an experiment to see what it takes to let my “AccessData” (server-side data processing) services work asynchronously. Turns out, not much on the spec side; what needs to be defined is:

- Async service discovery
- UWS details

2. Discovery in Datalink

Declare together with sync endpoint?

- no repetition of metadata when parameters are identical (as in TAP)
- either all AD services have to support async or there’d need to be some fancy technique to sense async capabilities
- => make async separate service in datalink definition. StandardIDs tell clients how to talk to them.

3. Datalink Example

```
<RESOURCE ID="apptmeh" type="meta" utype="adhoc:service">
  <GROUP name="input">
    <PARAM datatype="float" name="DEC_MIN" ucd="par.min;pos.eq.dec" unit="deg" value="">
      <DESCRIPTION>The latitude coordinate, lower limit</DESCRIPTION>
      <VALUES>
        <MIN value="15.9438963186"/>
        <MAX value="15.9638963473"/>
      </VALUES>
    </PARAM>
    <PARAM datatype="float" name="DEC_MAX" ucd="par.max;pos.eq.dec" unit="deg" value="">
    <PARAM arraysize="*" datatype="char" name="standardID" value="ivo://ivoa.net/std/SSDP#async"/>
  </GROUP>
  <GROUP name="output">
    <PARAM arraysize="*" datatype="char" name="standardID" value="ivo://ivoa.net/std/SSDP#sync"/>
  </GROUP>
</RESOURCE>
```

Full record

Known bug alert: The published service currently produces identical IDs for the two service declarations. A fix is being brought online RSN.

4. Service Operation

No excitement, plain UWS just works.

For want of an actual client, try:
```
```
(you can operate the job with your browser from http://dc.g-vo.org/califa/q2/dl/dlasync).