

## Finding VOEvent resources

Matthew J. Graham and Roy Williams  
*California Institute of Technology*

The VO registry infrastructure can be used to allow publication, discovery, and utilization of VOEvent resources. In particular, two types of resource specific to VOEvent are defined as registry schema: VOEventStream and VOEventServer. The stream schema describes a scientifically coherent collection of events that come from the same motivation, team, project, or experiment. Each event of a stream uses the same vocabulary (parameters) to describe events. The server schema describes which computers and interfaces can be used to send out future events (subscription capability) and/or can run queries on past events (query capability). It describes the streams that it knows, and then the capabilities it offers for each of those streams.

### Introduction

Although VOEvent is, by its nature, a very ephemeral construct, much of its associated infrastructure concerned with the generation, dissemination and persistence of packets is not. Working with VOEvent then involves working with the components of the VOEvent system but how does one discover what and where these are, how to work with them and what they are capable of?

The VO registry [1] is essentially a Yellow Pages for astronomical resources. Anything that is describable and identifiable – a resource – can be stored: not just data and services but also organizations, projects, software, etc. The registry holds a list of resource descriptions which are expressed as structured metadata, enabling automated processing and searching. Different types of resources conform to specific schema for that resource type.

The VOEvent infrastructure can be seen to be composed of two such resource types (defined as schema) [2]: **VOEventStream**, which encapsulates a set of events with a common provenance, and **VOEventServer**, which provides access points for getting event data in various ways.

### Using the registry for VOEvents

The type of activities that are supported by having VOEvent resources in the registry include:

- Resolving an event: the unique identifier for an event is a concatenation of a stream identifier and an event number within that stream. A question to the registry finds which VOEventServer has the query capability for that stream, and then that server can be probed to get the full event metadata.
- What is this event: the event identifier contains the VOEventStream identifier, from which the registry can produce the descriptive text, contact information for the responsible people, and so on.
- Subscription to a stream: VOEventServers are being built to push events to those who want them by a variety of messaging protocols; in order for an astronomer to set this up, the registry is asked which servers provide the "subscription capability" for the given stream, so that the astronomer can work with that server to set up future delivery of events.
- Creating a stream: A survey or project may wish to publish an event stream. The registry will provide a way to define the nature of the coming events (VOEventStream) and where to get them (VOEventServer), and thus allow others to find and subscribe to those events.
- What does this parameter mean: Each event has a set of keyword parameters; the registry can be used to find out their meanings.

## Identifiers and VOEvents

Every resource in the VO has a globally unique identifier (known as an IVORN - International Virtual Observatory Resource Name) [3], which provides a means for retrieving the metadata record for a specific resource from any registry. These identifiers begin with the scheme “ivo:” and have a modular syntax affording the control of the relevant parts to the appropriate entities:

```
ivo://my.datacenter.here/resourceName#localName
```

By splitting this into pieces, we can find other IVORNs buried inside. As with any URI, there can only be 0 or 1 pound signs (#) in the IVORN. If it is present, the `localName` (after the #) is not understood by the registry system, i.e. it is not retrievable. The part before the # (or the whole IVORN if there is no # present), however, is a reference to a retrievable resource record within the registry system. It is composed of the *Authority ID* (`ivo://my.datacenter.here`), which establishes a provenance for a set of particular resources, such as the holdings of an astronomy data center, and the *Resource Key* (`resourceName`), which identifies a particular resource within the set.

VOEvents from a particular source are not registered individually in the registry but rather as the coherent collection that is a VOEventStream. Each VOEvent can then be thought of as a representative of a class of events (the stream). The syntax for the IVORN of a VOEvent is:

```
ivo://my.datacenter.here/streamName#eventName
```

By splitting this on the # sign, we can look up the stream in the registry and find the meaning of parameters used in events from this stream. We can also discover which resources (VOEventServers) are handling this stream, either where can we subscribe for future events or where can we access stored past events.

## The VOEventStream Metadata Record

The resource description for a VOEventStream uses the common metadata of all VO resources [4] and also metadata that is peculiar to a VOEvent stream. Among the former will be:

- Identifier: the IVORN of the stream
- Title and short name: the formal name of the stream (e.g. the Catalina Real-time Transient Survey) and an abbreviated reference for it (e.g. CRTS)
- Curation: information about the publisher and creator of the stream with contact information
- Content: the nature of the stream’s content as a set of subject categories (e.g. radio astronomy, gamma-ray bursts) as well as a text description of it and a reference URL to relevant web pages (e.g. the survey website)
- Coverage: the sky coverage of the stream with the simplest representation being “AllSky”, or another being a cone of the sky. The temporal coverage of the stream can also be expressed here listing both previous and expected observing periods. Spectral coverage can also be expressed here.

## Parameter templates

A key feature of the VOEventStream concept is that each stream has a defined set of named “parameters” and a member event of a stream should only use parameters that are part of the specified list for that stream. While a VOEvent will specify a value of the correct data type for a parameter, the registered definition of the parameter will be valueless. Thus a ParamTemplate in a VOEventStream record would look like:

```
<Param name="max" ucd="meta.number" dataType="int">  
  <Description>Maximal number of records to retrieve  
  </Description>  
</Param>
```

but all that is needed in the event itself is the name-value association:

```
<Param name="max" value="50"/>
```

All of the other information can be added if the author desires so that the VOEvent can stand by itself. But when the event stream has been properly registered then only the name and value are strictly necessary.

In VOEvent, parameters can be collected into groups: for example, a measurement and its error estimate can be combined:

```
<Group name="Rmag">
  <Param name="value" ucd="phot.mag;em.opt.R" dataType="float"
    value="17.3"/>
  <Param name="error" ucd="phot.mag;em.opt.R;stat.err"
    dataType="float" value="0.1"/>
</Group>
```

The same information can be identically described in a VOEventStream record just by omitting the actual values. Note also that VOEvent and VOEventStream cannot have Groups within Groups: a Group may **only** contain Params and not other Groups.

The same rules of uniqueness are also in force for ParamTemplate and Groups in VOEventStream as for Params and Groups in VOEvent:

- Each Param must have a name. A Group without a name is equivalent to having a name which is the null string.
- Names must be unique within the set of Params not in a Group.
- Names must be unique for the set of Params within a given Group.
- Groups must have unique names – this means that only one Group can be nameless.

## The VOEventServer metadata record

Once the stream is defined in the registry, we want to know what kinds of services are available to deal with it. These are known as **Capabilities** in the VO registry model. With VOEventServer, there are three kinds of VOEvent-specific capability that we can describe:

- **subscription**: a client can submit a criterion, e.g. R magnitude brighter than 17, and events will be sent by the server in the future which satisfy that criterion.
- **formal query**: this is the Simple Event Access Protocol (SEAP, [5]), a formal request-response protocol for querying VOEvents.
- **informal query**: a web page with forms by which a collection of events can be browsed in some way.

The capability of a server is split further into **Interfaces**, which express precisely how a client can access the capability. For example, the subscription capability could be achieved by having the event sent out by email, instant messaging, or some bespoke messaging protocol. These are three different interfaces to the subscription capability and each will have different metadata associated with it (e.g. email address vs. phone number).

As with VOEventStream above, a resource description for a VOEventServer uses the common metadata of all VO resources [4] and also metadata that is peculiar to a VOEvent server. The first case of the latter is a list of streams to which the server can provide access through one of its capabilities. The details of the various capabilities and their respective interfaces are then described.

## Capabilities and interfaces

Although a VOEventServer can have any number of capabilities, two that are defined specifically to support the VOEvent infrastructure are SimpleEventAccess and Subscription.

The SimpleEventAccess defines metadata relating to the characteristics of a SEAP access point that the server might provide, such as the maximum query region size that is supported and a test query that can be used to ensure that the SEAP service is functional. Since the SEAP uses a straightforward HTTP endpoint, the details of the interface can be described using common VO resource metadata.

```

<capability xsi:type="voe:SimpleEventAccess"
  standardID="ivo://ivoa.net/std/VOEventSEAP">
  <interface xsi:type="vs:ParamHTTP" role="std">
    <accessURL use="base">
      http://nvo.caltech.edu/voeventnet/seap</accessURL>
    </interface>
    <maxQueryRegionSize>
    <maxRecords>10000</maxRecords>
    <testQuery>
      <seap:pos>120,15<seap:pos>
      <seap:size>0.5<seap:size>
    </testQuery>
  </capability>

```

The Subscription gives details of the various ways in which a client can receive future events, most of which information resides in the different interfaces under this capability. Specific metadata is defined in these interfaces for those access protocols peculiar in VO usage to VOEvent: Jabber, TCPV and RSS.

```

<capability xsi:type="voe:Subscription"
  standardID="ivo://ivoa.net/std/VOEventSubscribe">
  <interface xsi:type="voe:Jabber">
    <accessURL>http://moriori.cacr.caltech.edu</accessURL>
    <securityMethod standardID="ivo://ivoa.net/std/UserPass"/>
    <feedNode streamId="ivo://nvo.caltech.voeventnet/catot">
      home/moriori.cacr.caltech.edu/catalina</feedNode>
    </interface>
  <interface xsi:type="voe:RSS">
    <accessURL>http://www.voeventnet.org</accessURL>
    <feed streamId="ivo://nvo.caltech.voeventnet/catot">
      http://www.voeventnet.org/feeds/Catfeed.xml</feed>
    </interface>
  <supportsFilters>true</supportsFilters>
</capability>

```

## References

- [1] Graham, M.J., 2008, AN, 329, 304
- [2] Graham, M.J., Williams, R.D., Allan, A., Plante, R., Seaman, R., *VOEventServer: an XML Encoding Schema for Resource Metadata for Collection of Events*, v0.3, 2010  
<http://www.ivoa.net/internal/IVOA/IvoaVOEvent/VOEventServer-0.3.html>
- [3] Plante, R., Linde, T., Williams, R., Noddle, K., *IVOA Identifiers*, v1.12, 2007  
<http://www.ivoa.net/Documents/latest/IDs.html>
- [4] Plante, R., Benson, K., Graham, M.J., Greene, G., Harrison, P., Lemson, G., Linde, T., Rixon, G., Stebe, A., the IVOA Registry Working Group, *VOResource: an XML Encoding Schema for Resource Metadata*, v1.03, 2008  
<http://www.ivoa.net/Documents/latest/VOResource.html>
- [5] Graham, M.J., Allan, A., Seaman, R., Williams, R., Auden, E., Denny, B., Rots, A., Warner, P., *Simple Event Access Protocol (SEAP)*, v0.5, 2009  
<http://www.ivoa.net/Documents/latest/SEAP.html>