

Transient Astronomy with ACP Scheduler

Robert B. Denny

DC-3 Dreams SP, Mesa, Arizona

Abstract: Dispatch scheduling is ideally suited for the observing of transient events. With the addition of a VOEvent receiver and observing request generator, a dispatch scheduler can respond in a timely way to transient events yet still provide optimized timing for acquiring data on non-time-sensitive objects. Also provided is an alerting service that sends VOEvents upon completion of follow-up observations done in response to a VOEvent, as well as dynamic web pages that provide access to follow-up data (images, logs, etc.).

This is an informal and unpublished engineering white paper. It is intended for electronic distribution as a PDF document, thus it contains embedded hyperlinks to references. Footnotes with the referenced hyperlink have been provided for users of printed copies.

Release 2.1 – March 6, 2009

© 2008-2009, Robert B. Denny, Mesa, AZ (unpublished)

1 Introduction

In recent years, astronomers have increasingly focused on transient events such as gamma ray bursts, supernovae, cataclysmic variable star outbursts, and microlensing events. In response, the astronomical community has established a network with which detections of transient events can be relayed to observatories for follow-up observing. The time-sensitivity of these events requires prompt response by observers. Email, pagers, RSS feeds, etc are not fast enough.

The [International Virtual Observatory Alliance \(IVOA\)](#)¹ has overseen the construction of a system called [VOEvent](#)² for the rapid dissemination of information about transient events. The system uses [Extensible Markup Language](#)³ (XML) messages that flow over a variety of transports, all of which use the Internet (TCP/IP) at their lowest level. XML schemata for VOEvent messages have been published and, to some degree, are in use. A [Global VOEvent Network \(GVN\)](#)⁴ has been formed, and is growing. See Figure 1 to the right.

[ACP Scheduler](#)⁵ is commercial off-the-shelf software that provides an [automated dispatch scheduling](#)⁶ and observing system for a wide variety of modest (mostly commercially produced) telescopes and imagers in use by both amateurs and educational institutions. Scheduler maintains a database of observing requests which includes

target and image data information. Each observing request also includes one or more *constraints* within which image data is to be acquired. After completing an observing request, Scheduler re-evaluates the entire request database and makes a decision on which request to do next. This allows a new request to be added during observing, at which time it will immediately compete for observing time.

An important advantage of dispatch scheduling is that the dispatcher always decides what's best to observe at a particular time, given sky conditions and observatory location. Thus, interruptions for high-priority observing do not result in the previous workflow simply resuming where it was interrupted (the night's remaining observing targets may have moved into unfavorable positions). Instead, a fresh look at the request database is taken, and the appropriate requests for that time are serviced.

These design aspects make ACP Scheduler well-suited to handle transient-event observing; therefore a VOEvent receiver/alerter was developed to allow Scheduler to respond to transient events, take appropriate data, then send a VOEvent alert for those observations. See Section 3 below.

¹ <http://www.ivoa.net/>

² <http://www.ivoa.net/twiki/bin/view/IVOA/IvoaVOEvent>

³ <http://www.w3.org/XML/>

⁴ <http://www.estar.org.uk/wiki/index.php/VOEvent>

⁵ <http://scheduler.dc3.com/>

⁶ <http://scheduler.dc3.com/Scheduler2006.pdf>

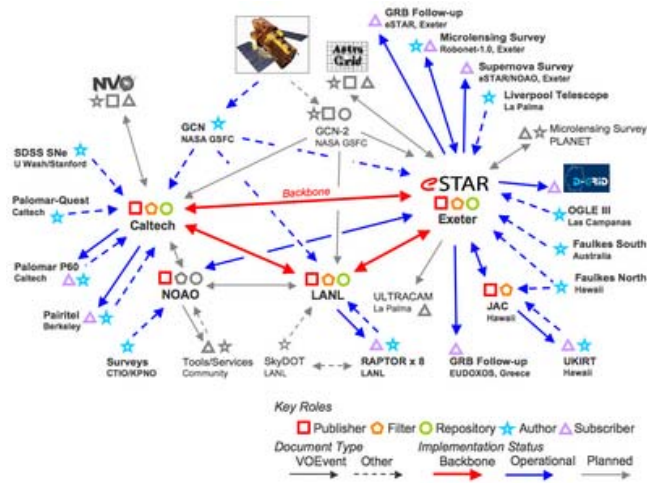


Figure 1. Global VOEvent Network

2 VOEvent Message Practicalities

The formally adopted XML schema set for VOEvent 1.1 messages includes an outer schema *VOEvent* plus an inner schema *Space-Time Coordinates* (STC). The VOEvent schema is straightforward and relatively simple. STC, however, is extremely complex. While it covers virtually any conceivable coordinate system, its complexity and its unsuitability as input to an automatic parser generator have prevented it from being implemented in its entirety.

VOEvent messages flowing on the GVN generally conform to the VOEvent schema down to the level of the *ObsDataLocation* node, where STC comes into effect. Within this sub-tree, an ad-hoc “simple” construction has come into use, and no published schema exists for it. It is approximately a sub-set of STC. Messages from different authors (originators) show variations that must be taken into account.

Finally, it appears that receivers on the GVN are lenient in accepting malformed XML. As a result, a significant fraction of the VOEvent messages on the GVN will not even load into some XML parsing engines (let alone validate against the referenced schemata) without being pre-patched to correct errors that appear in messages from various sources.

3 Overview: VOEvent Receiver/Alserter

In order to provide ACP Scheduler users with the ability to participate in transient astronomy, a combined VOEvent receiver and alserter was developed. Figure 2 below shows where the VOEvent receiver/alerter fits into the existing Scheduler observing system.

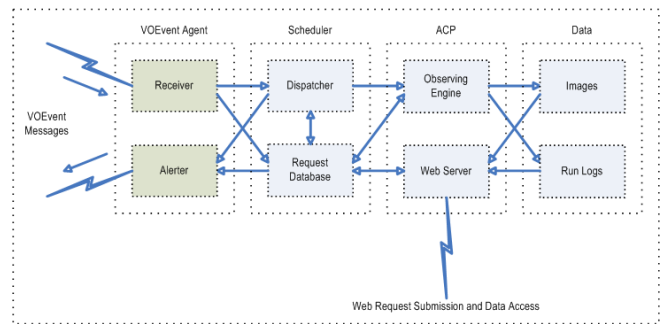


Figure 2. Overall system block diagram

VOEvent messages come into the receiver and result in observing requests being queued to the scheduler. If the event is time-critical, any in-progress observing is halted and the new request is started immediately.

Once the requested observations have been taken, the dispatcher signals the alserter, which sends a VOEvent message advising of the availability of the follow-up observation data. The follow-up alert message contains some information about the observatory as well as detailed information about each image that was acquired. Included in the latter are URLs for retrieving the FITS data.

4 The VOEvent Receiver

The role of the receiver is to receive VOEvent messages and, for those which are of interest, insert an appropriate observing request into the scheduler’s request database. Optionally, the receiver can interrupt the observing engine (if it is busy) thus forcing the dispatcher to immediately re-evaluate the pending requests and start the “best” one. If the event is time-critical, the dispatcher can be forced to start the new request immediately. Typical response times in that case are on the order of 10-20 seconds.

The receiver takes each incoming VOEvent message and processes it, either discarding it or using it to create an observing request for Scheduler. The processing steps are shown in Figure 3 below.

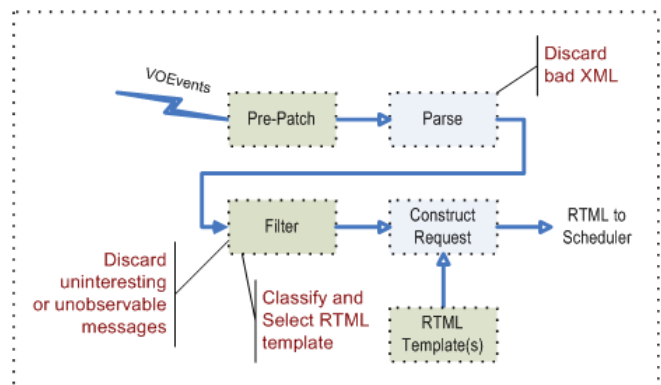


Figure 3. Receiver processing steps

4.1 VOEvent Source and Connection

As can be seen in Figure 1 above, the [eStar Project](http://www.estar.org.uk/wiki/index.php/Main_Page)⁷ at [University of Exeter \(UK\)](http://www.exeter.ac.uk/)⁸ is one of the two main hubs in the GVN, the other being the [VOEventNet](http://voeventnet.caltech.edu/index.html)⁹ at the [California Institute of Technology](http://www.caltech.edu/)¹⁰ (Caltech, USA). By default, the receiver gets its VOEvent messages from the [eStar event broker](http://www.estar.org.uk/wiki/index.php/ESTAR_Event_Broker)¹¹ at the. The receiver can alternatively be configured to receive its events from the Caltech event broker if desired.

Both the eStar and Caltech brokers provide a simple TCP socket service in which the receiver connects to a listener socket on the eStar broker. Once connected, the receiver will immediately get any VOEvent message coming from any source on the GVN. The connection remains open permanently. There is no polling by the receiver. Event messages arrive at the receiver within seconds of their arrival at the broker.

4.2 XML Pre-Patching

As outlined in section 2 above, some VOEvent messages contain non-compliant XML. In keeping with the philosophy of “be lenient in what you accept” from the internet, the receiver has a facility for pre-patching the message before trying to parse it (required to extract the data in the XML).

It is expected that the requirements for pre-patching will vary with time. Thus, in addition to the hard-coded pre-patching logic in the receiver itself (see 4.2.1 below), the receiver can use an external script or command line program to do any additional custom pre-patching. The raw message XML is piped into the script or program’s standard input and the patched XML is expected to be written to its standard output. The receiver simply waits for the external patcher to exit before moving to the next phase of processing. The pre-patched XML is then fed to the receiver’s VOEvent parser module, which contains its own pre-patching logic for the problems outlined later in this section.

A pre-patcher script can be in any scripting language that is supported on Windows, including Javascript, Perl, and Python. A command line executable program that serves as the patcher can be written in any language that generates a Windows executable and has access to standard in/out/err. It should be noted that the pre-patcher cannot use the Windows XML engine because it will reject malformed XML. Patching is best done using simple string and regular expression substitutions.

Finally, a special “null” custom pre-patcher script is supplied which simply archives all incoming VOEvent messages to disk. This is very useful when starting out, and may be useful for ongoing operations. If so, additional pre-

patching logic can be added to the message archiver so both tasks can be done.

4.2.1 Internally Patched VOEvent XML Problems

Tests performed in Spring 2008 on all of the VOEvent messages received by the eStar broker since 01-Jan-2007 revealed that the following items need pre-patching:

1. Root element missing namespace prefix; it must be `voe:VOEvent`.
2. Corresponding default namespace attribute must carry the `voe:` prefix.
3. The reference to the STC schema must be removed from the `ObsDataLocation` element.
4. Malformed `xsi:schemaLocation` attribute; schema name and location URL must be separated by whitespace
5. Illegal coordinate system value `FK5-UTC-GEO` in several places under `AstroCoordSystem`; it must be `UTC-FK5-GEO`.
6. Illegal use of `EventID` element, it must be `EventIVORN`.
7. Description element under `Why` element enclosed *within* a `Name` element. It must be a separate element under `Why`.
8. Param element containing ‘units’ attribute; the attribute must be ‘unit’ (singular).
9. `AstroCoordSystem` element containing ‘ID’ attribute; it must be ‘id’ (lower case’);
10. Inclusion of an `Instrument` element and sub-elements; these are not defined in the VOEvent schema.
11. Illegal value for the `ISOTime` element (form `yyyymmdd`); it must be a valid ISO 8601 time (form `yyyy-mm-ddThh:mm:ss`).

These items are pre-patched internally by the receiver’s VOEvent parser logic. Only items that arise in the future will need to be pre-patched by a script or program.

4.2.2 Pre-patching Considerations

It should be noted that the pre-patching process is fragile. Since the XML engine cannot be used for some of the patching operations, it is possible to incorrectly patch messages. Care must be taken to make the patching process as robust as possible (use of regular expression techniques is suggested).

4.3 VOEvent XML Parsing

After the incoming message has been cleaned up, it must be parsed in order to extract its data. This step is performed with internal logic using the standard Windows XML engine and object model code generated by the Liquid XML Data Binder. If the pre-patched message fails to parse due to remaining illegal XML, the failure details (including line and column numbers) are logged then the message is discarded.

⁷ http://www.estar.org.uk/wiki/index.php/Main_Page

⁸ <http://www.exeter.ac.uk/>

⁹ <http://voeventnet.caltech.edu/index.html>

¹⁰ <http://www.caltech.edu/>

¹¹ http://www.estar.org.uk/wiki/index.php/ESTAR_Event_Broker

A private schema is used to handle the known message constructions. In order to verify the suitability of this schema for parsing, a test was performed wherein all of the VOEvent messages received by the eStar broker since 01-Jan-2007 (13,000+) were passed through the pre-patcher and the parser. Parsing succeeded on all of the messages

If a message has one or more extraneous elements not covered by either the VOEvent or real STC schemas (and not already removed by pre-patching), they are simply ignored by the receiver's parser.

Due to the looseness of actual practice and the limits of the pre-patcher, it may be necessary to alter the parser in the future. In anticipation of this, the parser is an external component allowing easy replacement without new software releases.

4.4 Message Classification and Filtering

At this point, it is necessary to determine whether the incoming VOEvent message describes an event that is not only *interesting* but also *observable*. Only if both tests are passed will it trigger observing at the local observatory.

4.4.1 Interest Filter

Clearly, the decision as to whether the event is interesting is a subjective one. Therefore, the receiver uses an external script or program to examine the VOEvent message and make that determination. This affords the investigators maximum flexibility in that decision process.

The interest filter works similarly to the pre-patcher. The pre-patched and parsed message XML is piped into the script or program's standard input. The filter also receives the following command line arguments:

1. A list of "event type codes" for which the users are interested. This is used as a rough filter criterion; only certain event types are accepted. These codes are configurable in the receiver's settings.
2. The path to the directory containing the observing request RTML templates (see 4.5 below). This is used when the filter wants to construct the template *dynamically* based on the information in the received VOEvent.
3. Optional "signer id" if the incoming VOEvent was digitally signed (see section 7). The filter should check that this is equal to the Author IVORN in the message. If so, the message is assured to have been sent by that author.
4. Any additional arguments configured in the receiver's settings.

Once started, the filter must examine the received VOEvent message XML and determine whether the message should be acted upon. The standard supplied filter contains logic to process GRB alerts from the Swift satellite, the INTEGRAL satellite, and Catalina Sky Survey Transient alerts.

Since the XML has already been parsed internally by the receiver, it is virtually guaranteed to be usable in the Windows XML engine (which is accessible from Javascript and other scripting and programming languages). Therefore,

this task is best done via robust XML programming, using [XPath](#)¹² for accessing the data contained in the message.

Depending on the outcome of the filtering process, the filter returns a string containing up to four parts:

1. A name for the target, extracted from the VOEvent message data, or a blank string to indicate that the message is to be discarded.
2. An optional message to be displayed in a popup balloon on the user's desktop. This is most useful for alerting to certain discarded messages, since accepted ones always result in a popup.
3. For accepted messages, the name of an RTML template to be used to construct the observing request (see section 4.5 below).
4. An optional "flag". Currently only one flag "timecrit" is used. If it's present in the filter's return, the dispatcher is interrupted after the new observing request is entered.

The receiver simply waits for the external patcher to exit before moving to the next phase of processing. If the filter rejects the VOEvent, it should return only the popup message (if it wants the user to be informed of the discarded event) or nothing at all.

4.4.2 Observability Filter

Once a message has been determined to be interesting, the next step is to determine if the target is currently observable from the local observatory. This is done internal to the receiver, with help from the scheduler's sky-math library which is accessible via inter-process remoting of its API.

At this point, only a simple horizon test is made. The minimum altitude is a configuration setting of the receiver. As is described in 4.5 below, the observing request generated by the VOEvent may include additional constraints. If these are not met, the dispatcher will not start the event-generated observation, instead marking it as completed with error status (cannot meet constraints). If this happens, the dispatcher will simply resume its normal operation. Thus, there is another layer of observability criteria that may be applied.

4.5 Generating the Observing Request

ACP Scheduler supports a *variant* of the [Remote Telescope Markup Language](#)¹³ as its observing request input. This variant, RTML 2.3, supports most of ACP Scheduler's features and its standard constraints. Once the VOEvent message has passed all of the previous tests, it is used as the data source for generating an RTML observing request appropriate for the event. An RTML observing request can consist of multiple target coordinates, each of which can have a set of images in multiple color bands.

¹² <http://www.w3.org/TR/xpath>

¹³ <http://www.uni-sw.gwdg.de/~hessman/RTML/>

4.5.1 RTML Template Selection

As described in 4.4.1 above, the filter returns (among other things) the name of the RTML template to use for constructing the observing requests. The standard configuration includes templates for GRB alerts from Swift and INTEGRAL, as well as for the Catalina Sky Survey Transients. Thus, unique observing strategies can be used for each type of event.

4.5.2 Request Generation

Once the template has been selected, the final RTML for the observing request is generated and submitted. This is done by substituting data values from the VOEvent into the template then importing the resulting RTML into the scheduler's database. In the current implementation of the receiver, the following data is substituted into the template:

- Target right ascension (J2000)
- Target declination (J2000)
- Target name (from the interest filter, see section 4.4.1)
- Current time (used to force execution immediately)

It is anticipated that additional substitutions will be added as experience with the system is gained.

4.5.3 Dynamic RTML Generation

For advanced needs, the *filter* can construct an entire RTML request in a file, and then pass that file's name back as the template to use. If any of the substitution tokens appear in this dynamic RTML, substitution will be made, but in most cases the dynamically generated RTML will be ready to submit.

Thus, a sophisticated investigator could set the system up to examine a VOEvent in the interest filter, determine it is of interest, then extract data from the VOEvent, create a new RTML template just for that event, and return the RTML file name as the template to use, resulting in a custom "on the spot" observing plan for data acquisition. This scheme provides virtually unlimited flexibility in adapting observing to the VOEvent.

5 The VOEvent Alerter

The role of the alerter is to send out a VOEvent message announcing completion of follow-up observations that resulted from a previous incoming VOEvent. When the observations are complete, the scheduler signals the alerter which then inspects the completed observations in the database, as well as the data itself, and constructs the alert VOEvent message. See section A below for a sample of the alert message.

The format and content of the alert message is determined by code within the alerter and is not customizable. The alerter's settings do provide for input of the author information and observatory characteristics that are included in the message. In addition, a digital signature

can be applied to the outgoing message (see section 7). This is configurable in the alerter's settings.

The alerter sends the message using the simple TCP protocol to the Caltech [VOEventNet](#)⁹ broker. Currently, eStar does not support submission of VOEvent messages to their broker via simple TCP. This area of VOEvent is under development, and is expected to be discussed at the [Hot-wiring the Transient Universe 2](#)¹⁴ conference in April 2009.

6 Follow-up Data Access

The follow-up VOEvent message contains URLs to the data and to a dynamic web site that allows browsing all of the data from VOEvent follow-up observations. The web site is served by ACP's internal web server.

The web content is constructed automatically from the data (images, logs, saved VOEvent XML). Included are thumbnails of images, FITS header dumps, links to related VOEvent messages, and observing logs from ACP.

See Figure 4 and Figure 5 in section B below for sample screen captures. Note that the FITS header dump in Figure 5 continues below the bottom of the image.

7 Digital Signature Features

Both the receiver and the alerter support digital signatures for assuring message integrity and originator identity. The method used is described in the IVOA Note [A Proposal for Digital Signatures in VOEvent Messages](#)¹⁵. This method has not received formal approval by the IVOA. It is being implemented informally and will be discussed at the [Hot-wiring the Transient Universe 2](#)¹⁴ conference in April 2009. It is possible that digital signatures will be used to protect brokers using simple TCP from unauthorized submission of VOEvent messages.

If the required GnuPG software is installed on the system, if the relevant keys are available, and if enabled in the receiver's settings, the receiver will validate any digital signature on an incoming message. Depending on the receiver's settings, the message can be immediately discarded if the signature fails, or passed to the filter anyway. As previously described, the filter is responsible for matching the signer ID with the Author IVORN for the final authorship test.

If enabled in the alerter's settings, it will apply a digital signature to outgoing follow-up alert messages.

8 Status of Project

The ACP Scheduler VOEvent receiver as described herein is part of the production release 3.0 of ACP Scheduler.

¹⁴ <http://www.cacr.caltech.edu/hotwired2/>

¹⁵ <http://www.ivoa.net/Documents/latest/VOEventDigiSig.html>

9 References

- Denny, R: 2004, 2006 (unpublished) *Dispatch Scheduling of Automated Telescopes (rev.3, May, 2006)*
<http://scheduler.dc3.com/Scheduler2006.pdf>
- Rots, A: 2007 *Space-Time Coordinate Metadata for the Virtual Observatory*
<http://www.ivoa.net/Documents/latest/STC.html>
- Seaman, R., et al.: 2006 *Sky Event Reporting Metadata (VOEvent)*
<http://ivoa.net/Documents/latest/VOEvent.html>

10 Contact Information

Robert B. Denny
DC-3 Dreams, SP
rdenny@dc3.com
<http://dc3.com/>
+1 480 396 9700

Please feel free to contact the author for clarification and current status.

A. Sample of Alerter VOEvent Message

```
<?xml version="1.0" encoding="utf-8"?>
<voe:VOEvent xmlns:voe="http://www.ivoa.net/xml/VOEvent/v1.1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.ivoa.net/xml/VOEvent/v1.1 http://www.ivoa.net/xml/VOEvent/VOEvent-v1.1.xsd"
version="1.1" ivorn="ivo://com.dc3/engineering#0000000252" role="test">
  <Who>
    <AuthorIVORN>ivo://com.dc3/engineering</AuthorIVORN>
    <Date>2009-02-12T23:15:18</Date>
    <Author>
      <shortName>DC-3 Dreams, SP</shortName>
      <contactName>Bob Denny</contactName>
      <contactEmail>page-rdenny@dc3.com</contactEmail>
    </Author>
  </Who>
  <What>
    <Group name="Facility Data">
      <Param name="Aperture" ucd="instr.tel.aperture" value="60.0" unit="cm"/>
      <Param name="FocalLength" ucd="instr.tel.focalLength" value="600.0" unit="cm"/>
      <Param name="FieldOfViewX" ucd="instr.fov" value="28.5" unit="arcmin"/>
      <Param name="FieldOfViewY" ucd="instr.fov" value="20.2" unit="arcmin"/>
      <Param name="BasePixelScale" ucd="instr.pixel" value="0.83" unit="arcsec"/>
      <Param name="PlateCoordFrame" ucd="pos.frame" value="FK5"/>
      <Param name="DataAccess" ucd="meta.ref.url"
url="http://blaze.dc3.com/VOEvent/eventdata.asp?eid=0000000252&ivorn=ivo://nvo.caltech/voeventnet/catot#90128126036
4487642&date=2009-02-12T23:15:18"/>
      <Description format="text/plain">Multiply the BasePixelScale by the Binning of an image to get the pixel
scale of the image.</Description>
      <Description format="text/plain">Filter names are assigned by the facility. Contact the message Author for
details on filter characteristics.</Description>
      <Description format="text/plain">Data is protected by username/password. Contact the message Author for
access credentials.</Description>
    </Group>
    <Group name="Image 1">
      <Param name="JD" ucd="time.epoch" value="2454875.453646" unit="days"/>
      <Param name="Exposure" ucd="obs.exposure" value="300.0" unit="sec"/>
      <Param name="Filter" ucd="instr.filter" value="R"/>
      <Param name="Binning" ucd="instr.binning" value="1"/>
      <Param name="PlateRA" ucd="pos.eq.ra" value="59.669127" unit="deg"/>
      <Param name="PlateDec" ucd="pos.eq.dec" value="25.211336" unit="deg"/>
      <Param name="PlateCoordErrorRadius" ucd="pos.errorEllipse" value="0.4" unit="arcsec"/>
      <Param name="PlatePA" ucd="pos.posAng" value="359.99" unit="deg"/>
      <Param name="AirMass" ucd="obs.airMass" value="1.280"/>
      <Param name="Data" ucd="meta.ref.url" url="http://blaze.dc3.com/VOEvent/images/CATOT
#901281260364487642/CATOT #901281260364487642-Followup-001-R300-001-R.fts"/>
    </Group>
    <Group name="Image 2">
      <Param name="JD" ucd="time.epoch" value="2454875.457500" unit="days"/>
      <Param name="Exposure" ucd="obs.exposure" value="300.0" unit="sec"/>
      <Param name="Filter" ucd="instr.filter" value="R"/>
      <Param name="Binning" ucd="instr.binning" value="1"/>
      <Param name="PlateRA" ucd="pos.eq.ra" value="59.669121" unit="deg"/>
      <Param name="PlateDec" ucd="pos.eq.dec" value="25.211337" unit="deg"/>
      <Param name="PlateCoordErrorRadius" ucd="pos.errorEllipse" value="0.4" unit="arcsec"/>
      <Param name="PlatePA" ucd="pos.posAng" value="359.99" unit="deg"/>
      <Param name="AirMass" ucd="obs.airMass" value="1.260"/>
      <Param name="Data" ucd="meta.ref.url" url="http://blaze.dc3.com/VOEvent/images/CATOT
#901281260364487642/CATOT #901281260364487642-Followup-001-R300-002-R.fts"/>
    </Group>
    <Group name="Image 3">
      <Param name="JD" ucd="time.epoch" value="2454875.461481" unit="days"/>
      <Param name="Exposure" ucd="obs.exposure" value="300.0" unit="sec"/>
      <Param name="Filter" ucd="instr.filter" value="V"/>
      <Param name="Binning" ucd="instr.binning" value="1"/>
      <Param name="PlateRA" ucd="pos.eq.ra" value="59.669124" unit="deg"/>
      <Param name="PlateDec" ucd="pos.eq.dec" value="25.211334" unit="deg"/>
      <Param name="PlateCoordErrorRadius" ucd="pos.errorEllipse" value="0.4" unit="arcsec"/>
      <Param name="PlatePA" ucd="pos.posAng" value="359.99" unit="deg"/>
      <Param name="AirMass" ucd="obs.airMass" value="1.241"/>
      <Param name="Data" ucd="meta.ref.url" url="http://blaze.dc3.com/VOEvent/images/CATOT
#901281260364487642/CATOT #901281260364487642-Followup-001-V300-001-V.fts"/>
    </Group>
    <Group name="Image 4">
      <Param name="JD" ucd="time.epoch" value="2454875.465359" unit="days"/>
      <Param name="Exposure" ucd="obs.exposure" value="300.0" unit="sec"/>
      <Param name="Filter" ucd="instr.filter" value="V"/>
      <Param name="Binning" ucd="instr.binning" value="1"/>
      <Param name="PlateRA" ucd="pos.eq.ra" value="59.669121" unit="deg"/>
      <Param name="PlateDec" ucd="pos.eq.dec" value="25.211337" unit="deg"/>
    </Group>
  </What>
</VOEvent>
```

```

<Param name="PlateCoordErrorRadius" ucd="pos.errorEllipse" value="0.4" unit="arcsec"/>
<Param name="PlatePA" ucd="pos.posAng" value="359.99" unit="deg"/>
<Param name="AirMass" ucd="obs.airMass" value="1.223"/>
<Param name="Data" ucd="meta.ref.url" url="http://blaze.dc3.com/VOEvent/images/CATOT
#901281260364487642/CATOT #901281260364487642-Followup-001-V300-002-V.fts"/>
</Group>
</What>
<WhereWhen>
<ObsDataLocation xmlns="http://www.ivoa.net/xml/STC/stc-v1.30.xsd" xmlns:xlink="http://www.w3.org/1999/xlink">
<ObservatoryLocation>
<AstroCoordSystem id="UTC-GEOD-TOPO" xlink:href="ivo://STClib/CoordSys#UTC-GEOD-TOPO"/>
<AstroCoords coord_system_id="UTC-GEOD-TOPO">
<Position3D>
<Name1>Long</Name1>
<Name2>Lat</Name2>
<Name3>Elev</Name3>
<Value3>
<C1 pos_unit="deg">-111.6167</C1>
<C2 pos_unit="deg">33.7833</C2>
<C3 pos_unit="m">400.0</C3>
</Value3>
</Position3D>
</AstroCoords>
</ObservatoryLocation>
<ObservationLocation/>
</ObsDataLocation>
</WhereWhen>
<Citations>
<EventIVORN cite="followup">ivo://nvo.caltech/voeventnet/catot#901281260364487642</EventIVORN>
</Citations>
<Description format="text/plain">Followup observation(s) for ivo://nvo.caltech/voeventnet/catot#901281260364487642.
No measurements of target position or physical quantities have been made, refer to the What section for information and
location on the data. The data is made available for the investigators who issued the original VOEvent and any other
interested party. The URLs to data are protected by username/password.Contact the Author for access.</Description>
</voe:VOEvent>

```

B. Screen Captures of Follow-Up Data Web Pages

The screenshot shows a web browser window with the address bar containing the URL: `Followup Data for ivo://nvo.caltech/voeventnet/catot#901281260364487642`. The page header features the text "Red Mountain #663 | VOEvent Followup Data" in a dark blue bar. Below the header is a yellow navigation bar with the word "HOME" and a vertical line. The main content area is light blue and contains the following text:

For ivo://nvo.caltech/voeventnet/catot#901281260364487642
Original VOEvent message that triggered this observation
Followup VOEvent message sent to alert this observation
Automation system run log(s) for this observation
[20090212@231518.log](#)
[20090212@232537.log](#)

Below the text are four square thumbnails, each showing a star field. Each thumbnail is labeled with "Data details" in blue text and "300 sec. R" or "300 sec. V" below it. The first two thumbnails are labeled "300 sec. R" and the last two are labeled "300 sec. V".

Figure 4. Follow-up Web Page for a VOEvent



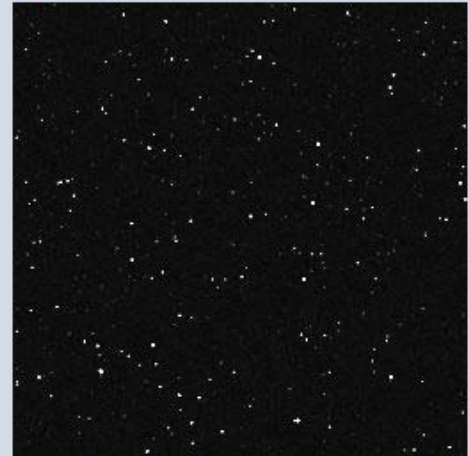
Red Mountain #663 | VOEvent Followup Data

HOME | FOLLOWUP DATA

CATOT #901281260364487642-Followup-001-R300-002-R

Data specifics:

Start time 2009-12-02 22:58:48 UTC
Integration time 300 sec.
Filter R
Center J2000 03:58:40.59 25d12m40.8s PA 360.0
Airmass 1.26
Detector geom. 1024 x 1024 pixels (binned 1x1)
Plate scales 1.63 x 1.63 arcsec/pixel
Field of view 27.90 x 27.90 arcmin



[Original FITS data](#) (or click on image)

Dump of FITS header:

```
Header listing for HDU #1:
SIMPLE = T / file does conform to FITS standard
BITPIX = 16 / number of bits per data pixel
NAXIS = 2 / number of data axes
NAXIS1 = 1024 / length of data axis 1
NAXIS2 = 1024 / length of data axis 2
COMMENT FITS (Flexible Image Transport System) format defined in Astronomy and
COMMENT Astrophysics Supplement Series v44/p363, v44/p371, v73/p359, v73/p365.
COMMENT Contact the NASA Science Office of Standards and Technology for the
COMMENT FITS Definition document #100 and other FITS information.
BZERO = 32768. / Unsigned 16-bit pixels
BSCALE = 1. / Unsigned 16-bit pixels
HISTORY File was created by PinPoint 5.0.14 at 2009-02-12T23:03:52
EXPOSURE= 3.000000000000E+002 / [sec] Duration of exposure
EXPTIME = 3.000000000000E+002 / [sec] Duration of exposure
DATE-OBS= '2009-02-12T22:58:48' / [ISO 8601] UTC date/time of exposure start
DATE = '12/02/09' / [old format] UTC date of exposure start
TIME-OBS= '22:58:48' / [old format] UTC time of exposure start
UT = '22:58:48' / [old format] UTC time of exposure start
TIMESYS = 'UTC' / Default time system
RADECSYS= 'FK5' / Equatorial coordinate system
XBINNING= 1 / Binning level along the X-axis
YBINNING= 1 / Binning level along the Y-axis
FILTER = 'R' / Filter name
OBSERVER= 'VOEvent' / Observer name
```

Figure 5. Web Page for Individual Image