

## Space Details

|                                   |                                                            |
|-----------------------------------|------------------------------------------------------------|
| <b>Key:</b>                       | AVED                                                       |
| <b>Name:</b>                      | 900260 Automated Visual Event Detection and Classification |
| <b>Description:</b>               |                                                            |
| <b>Creator (Creation Date):</b>   | brian (Jan 30, 2007)                                       |
| <b>Last Modifier (Mod. Date):</b> | dcline (Mar 31, 2010)                                      |

## Available Pages

- AVEDac 
  - AVED Beowulf Node Build Notes
  - AVED NFS Mounts and Storage Configuration
  - AVED XML Schema and Example
  - JAMSTEC - Toucan Laptop Installation Notes
  - UC Davis Video
- AVED Special Features
- AVEDac Install Howto
  - AVEDac Installation
    - AVED GUI Installation Notes ( Mac OS X )
    - AVEDac Beowulf Cluster Installation Notes
    - AVEDac Beowulf Cluster Miscellaneous Settings
    - AVEDac Beowulf Cluster MPICH Testing Notes
    - mbarivision Installation - Step 5. Build and install Berkeley MPEG Encoder (optional)
    - mbarivision Installation - Step 1. Build iLab Saliency Toolkit
    - mbarivision Installation - Step 2. Build and Install XML Libraries
    - mbarivision Installation - Step 3. Build and Install Transcode
    - mbarivision Installation - Step 4. Build and Install Quicktime or OpenQuicktime (optional)
    - mbarivision Installation - Step 6. Build and Install mbarivision or pmbarivision
- AVEDac Introduction
- AVEDac Support
- AVEDac FAQ
- AVEDac Running Howto
  - AVED Mbarivision Options
  - Installing AVED Scripts
- AVED Beowulf RAID administrative utilities

# Automated Visual Event Detection and Classification Overview

In order to study the distribution and abundance of oceanic animals, MBARI uses high-resolution video equipment on remotely operated vehicles. Quantitative video transects (QVTs) supplant traditional net tows to assess the quantity and diversity of organisms in the water column. QVTs are run from 50 m to 4000 m and provide high-resolution data at the scale of the individual animals as well as their natural aggregation patterns. However, the current, manual method of analyzing QVTs is labor intensive and tedious.

We are developing an automated system for detecting marine organisms visible in the video. Video frames are processed with a neuromorphic selective attention algorithm. The candidate objects of interest are tracked across video frames using linear Kalman filters. If objects can be tracked successfully over several frames, they are labeled as potentially "interesting" and marked in the video frames. The plan is that the system will enhance the productivity of human video annotators and/or cue a subsequent object classification module by marking candidate objects.

The continued use of ROVs and future use of Autonomous Underwater Vehicles (AUVs) for QVTs offer potential for even more data, perhaps many times what we current collect and analyze. Hence, we see tremendous benefit in automating portions of the analysis. We also see great benefit in automating analysis of video from fixed ocean observatory cameras, where autonomous response to potential events (pan/zoom to events), and automated processing of largely "boring" (event sparse) video streams from 10s or 100s or even 1000s of network cameras could be key to those cameras being useful practical scientific instruments.

[External Project Website](#)

**For developers and users - the code and documentation has been moved from Confluence to google code**

---

[AVEDac Google Code](#)

## For administrators

---

[Condor pool statistics](#)  
[AVED Beowulf RAID administrative utilities](#)  
[MBARI Beowulf cluster connection diagram](#)  
[MBARI cluster rack order diagram](#)  
[Beowulf node build notes](#)  
[Project NFS Mounts and Storage Configuration](#)  
[AVED XML Schema and Example](#)

# These are notes for building the MBARI Beowulf cluster nodes.

**Step 1. Unpack system, plug in keyboard, mouse, and compatible monitor.**

**Step 2. Turn on system and hit the DEL key to enter into BIOS.**

**Step 3. Under SATA Options in BIOS, change from "Enhanced" to "Compatible" hard disk options. This allows the Fedora installation to proceed. Without this, the install fails.**

**Step 4. Under Boot options in BIOS, change boot order to boot first from CD and then HD.**

**Step 5. Insert Fedora Core 3 (FC3) installation disk and reboot system.**

**Step 6. While system is booting, perform a CTL-S to view system information. Write down the system MAC address. Hit F4 to continue.**

**Step 7. Install FC3 in "Server" Mode from CD. Erase all system partitions, disable SELinux and firewall, and only install default server applications.**

**Step 8. Reboot again and change boot order to first boot from HD and then CD.**

**Step 9. Copy and unzip [this](#) to a USB thumbdrive.**

**Step 10. Install network driver from thumbdrive**

The Beowulf cluster has two different nodes; each group was purchased at different times and each group has a different network driver, thus the need for different drivers noted below.

| For nodes 1,5, and 8                                                                                                                                                                                                                                                                                     | For nodes 2,3,4,6 and 7                                                                                                                                                                                                                                              |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Add the following lines to /etc/modprobe.conf:<br><br>alias eth0 e1000<br>alias eth1 e1000<br><br>alias bond0 bonding<br>mount /media/usbdisk<br>cd /media/usbdisk/e1000-7.4.27/src<br>install -D -m 6644 e1000.ko /lib/<br>modules/2.6.9-1.667smp/kernel/drivers/net/<br>e1000/e1000.ko<br>/sbin/depmod | Add the following lines to /etc/modprobe.conf:<br><br>alias eth0 igb<br>alias eth1 igb<br><br>mount /media/usbdisk<br>cd /media/usbdisk/igb-2.1.9/src<br>install -D -m 6644 igb.ko /lib/<br>modules/2.6.9-1.667smp/kernel/drivers/net/igb/<br>igb.ko<br>/sbin/depmod |

**Step 11. Setup channel bonding, e.g. for node 1**

1. Configure the following files to support LAN redundancy. Create a bond0 file, ifcfg-bond0 in the /etc/sysconfig/network-scripts directory. Include the following information in the ifcfg-bond0 file:

```
DEVICE=bond0
IPADDR=192.168.1.1
NETMASK=255.255.255.0
NETWORK=192.168.1.0
BROADCAST=192.168.1.255
ONBOOT=yes
BOOTPROTO=none
USERCTL=no
```

2. Create an ifcfg-ethn file for each interface in the bond. All interfaces should have SLAVE and MASTER definitions. Edit the ifcfg-eth0 file to appear as follows:

```
DEVICE=eth0
USERCTL=no
ONBOOT=yes
MASTER=bond0
SLAVE=yes
BOOTPROTO=none
```

3. Edit the ifcfg-eth1 file to appear as follows:

```
DEVICE=eth1
USERCTL=no
ONBOOT=yes
MASTER=bond0
SLAVE=yes
BOOTPROTO=none
```

4. Add the following lines to /etc/modprobe.conf:

```
alias bond0 bonding
options bond0 miimon=100 mode=1
```

5. Restarting Networking

Restart the networking subsystem. From the console of either node in the cluster, execute the following command :

```
/etc/rc.d/init.d/network restart
```

## **Step 12. Change the hostname, e.g. for node 1**

```
hostname node1.private.net
```

## **Step 13. Modify the fstab file to include mounts in the fstab file on the thumbdrive**

## **Step 14. Replace the sshd\_config with the file on the thumbdrive**

```
cp -f /mount/usbdisk/sshd_config /etc/ssh
```

## AVED NFS Mounts and Storage Configuration

This page last changed on Dec 17, 2008 by [dcline](#).

### NFS mounts

Most AVED computers have the following mounts available that are used to share files around the MBARI network and between AVED computers.

| mount point  | description                                                                                                                                                                                                                              |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| /u           | It's a good idea to put your source code here because it is backed up hourly and daily. Maps to your tempest users root directory, e.g. /u/dcline, or in Windows IE navigate to \\tempest\dcline.                                        |
| /tempbox     | Can be used to share files between AVED computers and Windows or Mac machines. Maps to your temporary directory on tempest, e.g. /tempbox/dcline, or in Windows IE navigate to \\tempest\tempbox\dcline.                                 |
| /engineering | Included for reference. Engineering data was previously placed here, but is planned to be migrated to Confluence. Maps to the engineering directory on tornado, /tornado/engineering, or in Windows IE navigate to \\tornado\engineering |
| /project     | Included for reference. Engineering data used to be placed here but is slowly being migrated to Confluence. Maps to the project library, e.g. /tornado/projectlibrary, or in Windows IE navigate to \\tornado\projectlibrary             |

AVED specific shares available are:

| mount point  | description                                                                                                                                                                            |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| /nanomiaRAID | Maps to the RAID storage on nanomia for long-term data storage of video, in Windows IE navigate to \\nanomia\nanomiaRAID, on Mac, connect to smb://nanomia.shore.mbari.org/nanomiaRAID |

If these mounts are not on you Linux computer, you can add them if you have root permission to your /etc/fstab file with:

|                               |              |     |                           |   |
|-------------------------------|--------------|-----|---------------------------|---|
| tempest:/vol/vol0/users       | /u           | nfs | bg,soft,nosuid,nodev,exec | 0 |
| tempest:/vol/vol0/tempbox     | /tempbox     | nfs | bg,soft,nosuid,nodev,exec | 0 |
| tornado:/vol/vol0/Engineering | /engineering | nfs | bg,soft,nosuid,nodev,exec | 0 |
|                               | /project     | nfs | bg,soft,nosuid,nodev,exec | 0 |

|                                             |             |     |            |   |   |
|---------------------------------------------|-------------|-----|------------|---|---|
| tornado:/vol/vol0/ProjectLibrary            |             |     |            |   |   |
| nanomia.shore.mh/nanomiaRAID<br>nanomiaRAID | nanomiaRAID | nfs | rw,bg,soft | 0 | 0 |

**Storage Configuration**

## AVED XML Schema and Example

---

This page last changed on May 12, 2009 by [dcline](#).

### XML Schema

Main schema file: [EventDataSet.xsd](#)

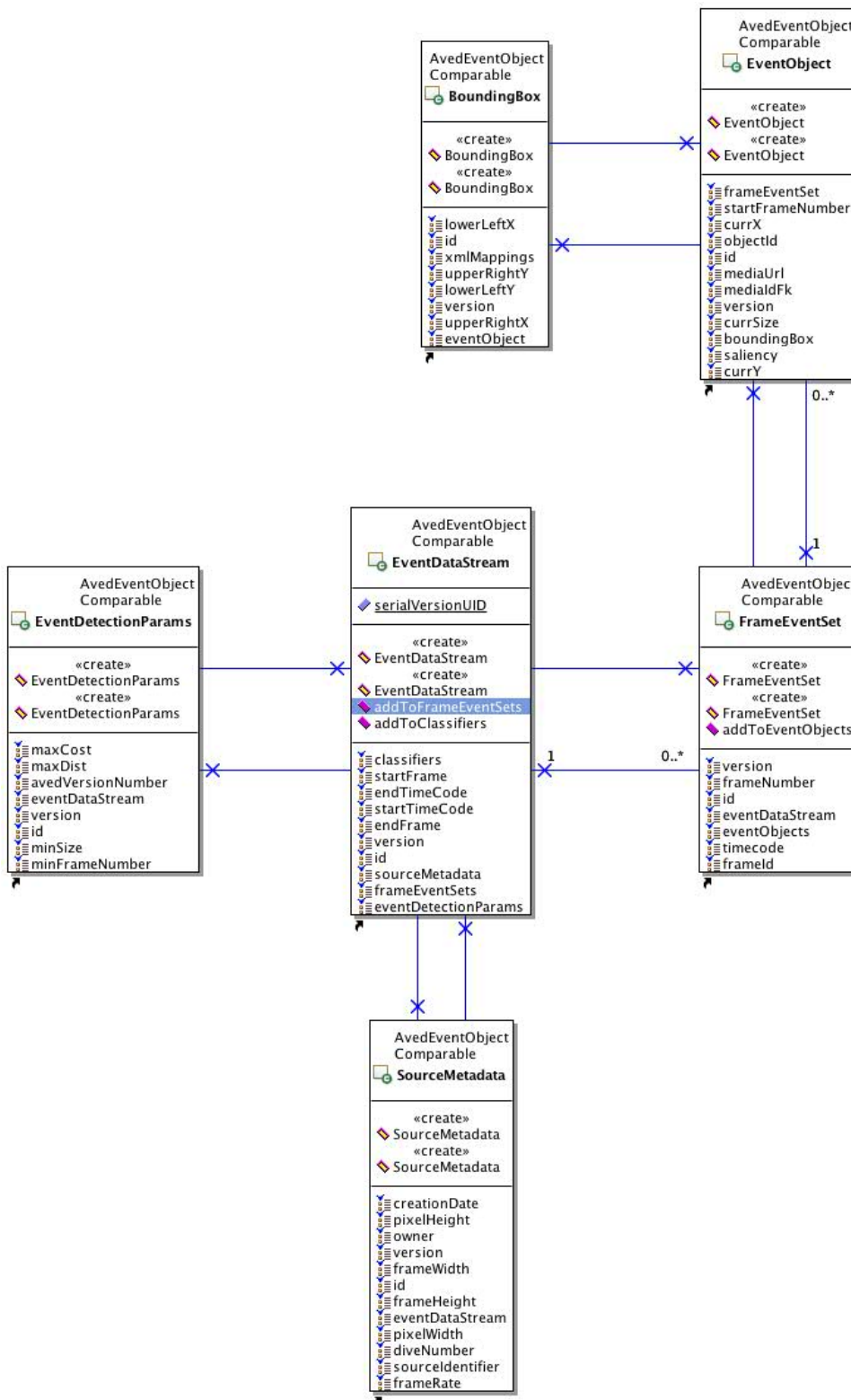
| -----> [SourceMetadata.xsd](#)  
| -----> [EventDetectionParameters.xsd](#)

### Data Model



#### Important

- AVED indexes output by *FrameNumber* by default. If the input video clip is encoded according to the [UTC](#) format, the timecode will be populated in the *FrameEventSet->timecode* field. Timecode frequency is calculated based on the *SourceMetaData->frameRate* field.



## XML Sample

```

<?xml version="1.0" encoding="UTF-8" ?>
<!--Created by: MBARI software mbarivision v0.1.3 (C) 2003-2004 MBARI built Sep 8 2006 at 13:58:18
-->
<EventDataSet
xmlns="http://www.w3.org/2001/XMLSchema-instance"
CreationDate="Sun Sep 24 14:15:05 2006&#xA;"

```



```

EndFrame="1776"
StartFrame="0"
StartTimeCode="">
<EventDetectionParameters
VersionNumber="mbarivision v0.1.3 (C) 2003-2004 MBARI built Sep 8 2006 at 13:58:18;"
maxCost="6885"
maxDist="18"
minFrameNum="5"
minSize="100"/>
<FrameEventSet FrameNumber="0" TimeCode=""/>
<!-- FrameEventSets omitted from sample for brevity -->
<FrameEventSet FrameNumber="143" TimeCode=""/>
<FrameEventSet FrameNumber="144" TimeCode="">
<EventObject CurrSize="2043" CurrX="378" CurrY="430" ObjectID="1" Saliency="0.00257649"
StartFrameNumber="140">
<BoundingBox LowerLeftX="342" LowerLeftY="453" UpperRightX="421" UpperRightY="411"/>
</EventObject>
</FrameEventSet>
<FrameEventSet FrameNumber="145" TimeCode="">
<EventObject CurrSize="1925" CurrX="376" CurrY="431" ObjectID="1" Saliency="0.00213244"
StartFrameNumber="140">
<BoundingBox LowerLeftX="336" LowerLeftY="452" UpperRightX="428" UpperRightY="412"/>
</EventObject>
</FrameEventSet>
<FrameEventSet FrameNumber="146" TimeCode="">
<EventObject CurrSize="2126" CurrX="376" CurrY="431" ObjectID="1" Saliency="0.00213244"
StartFrameNumber="140">
<BoundingBox LowerLeftX="338" LowerLeftY="454" UpperRightX="424" UpperRightY="413"/>
</EventObject>
</FrameEventSet>
<!-- FrameEventSets omitted from sample for brevity -->
<FrameEventSet FrameNumber="658" TimeCode=""/>
<FrameEventSet FrameNumber="659" TimeCode="">
<EventObject CurrSize="255" CurrX="261" CurrY="441" ObjectID="4" Saliency="0.00343171"
StartFrameNumber="655">
<BoundingBox LowerLeftX="245" LowerLeftY="448" UpperRightX="275" UpperRightY="436"/>
</EventObject>
</FrameEventSet>
<FrameEventSet FrameNumber="660" TimeCode=""/>
<!-- FrameEventSets omitted from sample for brevity -->
<FrameEventSet FrameNumber="1773" TimeCode=""/>
<FrameEventSet FrameNumber="1774" TimeCode=""/>
<FrameEventSet FrameNumber="1775" TimeCode=""/>
<FrameEventSet FrameNumber="1776" TimeCode=""/>
</EventDataSet>

```

## JAMSTEC - Toucan Laptop Installation Notes

---

This page last changed on Apr 22, 2010 by [dcline](#).

# Toucan

---

### About Toucan

Toucan is the name of the laptop for our JAMSTEC collaborator Dhugal Lindsay. It is intended to be a test platform for JAMSTEC to explore possible uses for AVED. The AVED software suite was installed on this computer by MBARI.

Toucan is a ThinkPad Lenovo T61p.

Fedora Core 8 was installed and configured for this laptop by the Emperor Linux Company <http://www.emperorlinux.com/>.

### MBARI Network Settings

Network settings **should be changed per JAMSTEC network configuration**.

Currently configured as:

IP: Dynamically assigned  
Primary DNS 1 134.89.12.72  
Submask 255.255.254.0  
Gateway 134.89.12.1  
Secondary DNS 2 134.89.12.86

To change these, select the menu Systems->Administration->Network, and change accordingly.

### AVED Installation

AVED was installed according to the instructions [AVEDac Installation](#). The AVED software and its dependencies were installed as the user **root**. Additionally, a user **aved** was installed to use. Passwords for both the root and aved users have been sent to Dhugal.

Two additional steps were required for this installation:

1. Add manual links for the dc1394 control libraries. This was needed to get saliency to compile.

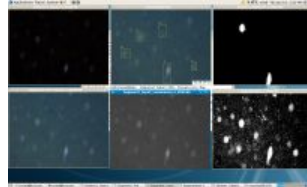
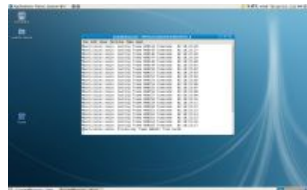
```
cd /usr/lib
ln -s libdc1394.so.22 libdc1394_control.so
ln -s libdc1394.so.22 libdc1394.so
```

2. Add /usr/local/lib to the /etc/ld.so.conf. This was needed to get the loader to load the dependent libraries (e.g. xercesc) that are by default installed to /usr/local/lib to load.

```
echo /usr/local/lib >> /etc/ld.so.conf
```

### JAMSTEC Video Data Locations and Processing Scripts

Three sets of test high-definition video data were installed on this laptop. Two are sets of still pictures, and one is a video clip. All of this data was provided by JAMSTEC on tape, or digitally. Some basic scripts were provided in addition to the AVED scripts, to give examples of how to process the data. These scripts are noted in the JAMSTEC Scripts table below. For more information on running AVED, please see the [AVEDac Running Howto](#) guide.

| Directory                                                                                                                                                                                                                                                                                                                                                                                                               | Data Type                                                                                                                  | Process Command                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| /home/aved/Pictures/benthic/<br>benthic_a                                                                                                                                                                                                                                                                                                                                                                               | Individual ppm frames from (I think) a PICASSO benthic video recording. Frames were extracted from HD tape.                | <p>cd /home/aved/Pictures/benthic/<br/>benthic_a<br/>jamstec_runmbarivis_displayalg<br/>This will display output, similar to the following. *NOTE* Windows display overlapped, not in the cascade pattern shown in this image. Just remember to click and drag them from on top of each other.<br/>(click to enlarge)</p>  <p>OR<br/>jamstec_runmbarivis<br/>This will simply spew out the output from mbarivision into the console window.<br/>(click to enlarge)</p>  |
| /home/aved/Pictures/midwater/<br>midwater/midwater_fasttransect                                                                                                                                                                                                                                                                                                                                                         | Individual ppm frames from (I think) a PICASSO fast moving midwater video recording. Frames were extracted from a HD tape. | <p>cd /home/aved/Pictures/<br/>midwater/midwater/<br/>midwater_fasttransect<br/>jamstec_runmbarivis or<br/>jamstec_runmbarivis_displayalg</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| /home/aved/Pictures/<br>midwater/midwater/<br>midwater_slowtransect                                                                                                                                                                                                                                                                                                                                                     | Individual ppm frames from (I think) a PICASSO slow moving midwater video recording. Frames were extracted from a HD tape. | <p>cd /home/aved/Pictures/<br/>midwater/midwater/<br/>midwater_slowtransect;<br/>jamstec_runmbarivis or<br/>jamstec_runmbarivis_displayalg</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| /home/aved/<br>Video/20080604T063139Z.mov<br><br><div style="background-color: #e0ffe0; padding: 5px; border: 1px solid #c0ffc0;">  <b>Handy Hint</b><br/> This file was named according to the <a href="#">ISO8601</a> date and time convention as the ISO standard is used by AVED scripts to timestamp events by timecode, </div> | Quicktime movie encoded with mpeg4 codec. Source was received via FTP from Dhugal.                                         | <p>cd /home/aved/<br/>Video/20080604T063139Z.mov;<br/>jamstec_runprocess<br/>20080604T063139Z.mov</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |

|                                                                                                                                     |  |  |
|-------------------------------------------------------------------------------------------------------------------------------------|--|--|
| <p>instead of frame number.<br/>E.g.<br/>20080604T063139Z<br/>assumes the tape recording started on 06/06/2008 at 06:31:39 UTC.</p> |  |  |
|-------------------------------------------------------------------------------------------------------------------------------------|--|--|

## JAMSTEC Scripts

The following are some simple scripts to help get you started:

| Script                                    | Description                                                                                                                                                                                                                                                                                                                               |
|-------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| /home/aved/jamstec_runmbarivis            | Simple script that clears the MBARIVISION_OPTIONS variable and runs the mbarivis_command script. Nothing special here - just a wrapper script. Use this to process the benthic_a or midwater_fast/slowtransect images. <b>This must be executed in the root directory of the ppm frames</b> , e.g. /home/aved/Pictures/benthic/benthic_a. |
| /home/aved/jamstec_runmbarivis_displayalg | Simple script that sets the mbarivision options to display the AVED algorithm output at various stages. <b>This must be executed in the root directory of the ppm frames</b> , Also rescales the input to 640x480 to make the display more manageable.                                                                                    |
| /home/aved/jamstec_runprocess             | Simple wrapper script that executes the runclip AVED script, and displays the output in the vlc viewer. Only processes the first 100 frames of the clip, so remove the argument that specifies the frame range, and by default this will process the entire clip.                                                                         |

This page last changed on Jun 09, 2010 by [dcline](#).

## Requirements for AVED processing

AVED is designed to process individual images, therefore video must be broken apart into individual image "frames" before processing.

## About Transcoding and Transcode

Transcoding is the digital-to-digital conversion of one encoding to another.

The provided MBARI scripts use a tool called [transcode](#) to convert video into to individual PPM for processing. Therefore, for this project, it is required to convert video into a a format transcode can convert into into PPM frames.

## UC Davis Video Specification

UC Davis cameras produce video that will need to be converted into a format supported by [transcode](#)

UC Davis camera video format :

track 0: 640x480 29.97 fps 1953.70 kbits/sec data rate

track 1: audio, Stereo, 48.000 KHz



### Time code support

If time code support is needed, we'll need to install the MBARI custom transcode and need to record ISO8601 convention (see below). This would record the events by time code, instead of by frame number. This might be useful to referencing with ancillary data, for example.

IS8601 File naming convention

To support timecode, files need to be named according the UTC time of the first frame following the [ISO8601](#) conventions. For example:

**20080607T120256.mov, 20080607T120256.avi.** For example, **20080607T120256.mov** represents a clip recorded on June 7th, 2005 (06-07-2008), and the recording started at 12:02:56 UTC.

# Installing an AVED-enabled Condor node

## RPM installation

- As root install condor RPM

```
rpm -ivh condor-xxxxx-linux-x86-rhel3-dynamic-1.i386.rpm
```



**error: Failed dependencies: libstdc++.so.5 is needed by condor-xxxxxxxxx**  
Install libstdc++.so.5: yum -y install compact-libstdc++-33

- Condor require the CONDOR\_CONFIG variable sets to the emplacement of its config file
- Edit /opt/condor-xxxxxx/etc/condor\_config changing the lines to allow READ/WRITE access and an address where email should be sent when something goes wrong

```
HOSTALLOW_WRITE = *.your.domain  
CONDOR_ADMIN = <your email here>
```

- Run configure script to setup installation as a submit and execute only node

```
/opt/condor-xxxxxx/condor_configure --type=submit,execute,manager --owner=user --install-dir=/opt/condor-xxxxxx
```



Make sure in /opt/condor-xxxxxx/etc/condor\_config line reads: DAEMON\_LIST = COLLECTOR, MASTER, NEGOTIATOR, SCHEDD, STARTD

## Installing Condor as a service on Linux

- Create condor.sh file in /etc/profile.d then add the following to it:

```
CONDOR_ROOT=/opt/condor-xxxxxxxxxx  
export PATH=$PATH:$CONDOR_ROOT/sbin:$CONDOR_ROOT/bin  
export CONDOR_CONFIG=$CONDOR_ROOT/etc/condor_config
```

- Install condor service script to execute condor on bootup

```
cd /opt/condor-xxxxxx/etc/examples  
cp condor.boot /etc/init.d/condor
```

- Edit condor service script replacing MASTER line

```
MASTER=$CONDOR_ROOT/sbin/condor_master
```

- Add . /etc/profile.d/condor.sh **before** the line MASTER

```
. /etc/profile.d/condor.sh  
MASTER=$CONDOR_ROOT/sbin/condor_master
```

- Add soft links to the condor startup script

```
ln -s /etc/init.d/condor /etc/rc2.d/S100condor  
ln -s /etc/init.d/condor /etc/rc2.d/K100condor
```

- Start the condor service

```
service condor start
```

## Configuring Condor

Change the following value in the `/opt/condor-xxxxxxx/local.aaaaaa/condor_config.local`

```
## When is this machine willing to start a job?
START = TRUE

## When to suspend a job?
SUSPEND = FALSE

## When to nicely stop a job?
## (as opposed to killing it instantaneously)
PREEMPT = FALSE

## When to instantaneously kill a preempting job
## (e.g. if a job is in the pre-empting stage for too long)
KILL = FALSE
```

## Installing AVED as a web-service

A web-service has been written to make the system easy to interface and automate. AVED web-service requires AVED and its dependencies to be installed (cf. AVED Guide section Installation) and requires Condor as well (cf. AVED Guide section Special Features-Installing an AVED-enabled Condor node). Moreover, in order to deploy AVED web-service, a web application server is required. We'll consider in this Guide starting from scratch, so if you already have an application server running and install go to section: Deploy the AVED web-service.

The AVED web-service is distributed as source files that you must compile yourself, however, an ant script is provided to help you in this step. Here is a short overview of the steps to build and deploy the AVED web-service from sources:

- **Preparation:** AVED web-services required a web server and ant to be built. See Installing AVED as a web-service - Preparation section for more information.
- **Download** the AVED source files (distributed as tar'ed and compressed files). Place the file in a place of your own choice.
- **Untar and uncompress** the file to the place you want the program installed. You will then need to read the next section about how to configure before you compile the program. Below is shown the exact commands: `tar -xzf /path/to/aved-web-service-1.0.0.tar.gz`
- Now change to the new directory  
`cd aved-web-service`
- Run ant.  
`ant`
- copy and paste the build/AvedWebService.aar files into the `<path-to-the-web-server>/webapps/axis2/WEB-INF/services`  
`cp build/AvedWebService.aar <path-to-the-web-server>/webapps/axis2/WEB-INF/services`
- copy and paste the jars needed by the web service from the jars folder into the `<path-to-the-web-server>/webapps/axis2/WEB-INF/lib`  
`cp jars/*.jar <path-to-the-web-server>/webapps/axis2/WEB-INF/lib`
- restart your web server application

### Preparation for deploy the AVED web-service

In this section we're gonna install the application server Apache Tomcat, the web-service utils with Apache Axis2, and finally ant to be able to build the AVED web-service. If you already have an application web-server installed and running you can directly go to the Deploy the AVED web-service.

## Installing Java

Apache Tomcat and ant require Java to be install on your computer. You can process different way: RPMs, self-extracted, or sources. If you're using sources, download the binaries distribution from <http://www.sun.com/java/>. Then untar and set the JAVA\_HOME environmental variable to the full path of your Java distribution.

```
tar xzvf jdkx.xxx.tar.gz
export JAVA_HOME=<full-path-to-Java>
```

## Installing apache Tomcat

To install Apache tomcat, first download it from <http://tomcat.apache.org/> (download the core version). Then extract it into the folder where you want to setup your application server.

```
tar xzvf apache-tomcat-xxxxxx.tar.gz
```

To test if your application server is on, just start it, you have to be in the apache-tomcat folder:

```
./bin/catalina.sh start
```

Start your favorite browser and tape: <http://localhost:8080/>, if everything is working you should fall on the apache-tomcat presentation page. Now your application server is up, download the web-service utilities from <http://ws.apache.org/axis2/> (download the war distribution) and copy it into the <full-path-to-tomcat>/webapps folder.

```
cp <path-to-axis2>/axis2.war <path-to-tomcat>/webapps
```

Then on your browser, tape: <http://localhost:8080/axis2>. You should find at this URL the axis2 presentation page.

## Installing apache Ant

The binarie distribution can be found at: <http://ant.apache.org/bindownload.cgi>. Download it and untar it into the folder you want ant to be install. You can install it from rpms as well. Ant is not required to build the AVED web-service, it's just more convinient to build it, but you can compile it by yourself using javac. Don't forget to add to the classpath the jars included into the aved-webservice/jars folder.

```
tar xzvf apache-antxxxxxxx.tar.gz
```

## Allowing condor to be requested from http

Open the file <path-to-condor>/etc/condor\_config and add the following lines (wherever in part 1)

```
SCHEDD_ARGS=-p 8181
ENABLE_SOAP = TRUE
ALLOW_SOAP = */*
ENABLE_WEB_SERVER = TRUE
QUEUE_ALL_USERS_TRUSTED = TRUE
```

You can set the SCHEDD\_ARGS to the port you prefere, here we choose 8181. You have to restart condor or juste reconfig it:

```
condor_reconfig
```

## Deploy the AVED web-service

To deploy the AVED web-service, ant and an application web-server are required, if you don't have any please refer to AVED Guide - Special Features section Installing AVED as a web-service - Preparation for deploy the AVED web-service.

The web service provided condains the service (src/org/mbari/aved/webservice/server/AvedService.java) and an exemple to request it (src/org/mbari/aved/webservice/client/AvedClient.java). Before deploying the web-service, you have to edit the java code in order to provide the user Condor is gonna run with and the computer sets. For that edit the file AvedService.java in src/org/mbari/aved/webservice/server



```
public static String SCHEDD_LOCATION = "http://host.domain.name.org:8181";
public static String USER = "aved";
```

Then, compile the source by tapping

```
ant
```

If everything is going well, you should see a BUILD SUCCESSFUL printed, and a build folder should have been created. In these folder you'll find a AvedService.aar file what is the web-service ! To deploy it, just copy and past this file into your application server directory, and the jars from the jars into the application server lib directory:

```
cp build/AvedService.aar <full-path-to-tomcat>/webapps/axis2/WEB-INF/services/
cp jars/*.jar <full-path-to-tomcat>/webapps/axis2/WEB-INF/lib/
```

Restart tomcat, and check with your browser if your web-service is up by tapping <http://host:8080/axis2/services/listServices>, AvedService should be listed.

The AVED web-service requires 4 envirenmental variables:

```
export MBARIVISIONROOT=<full-path-to-mbarivision>
export JAVA_HOME=<full-path-to-Java>
export CONDOR_CONFIG=/opt/condor-xxxxxxx/etc/condor_config
export CONDOR_SPOOL=/opt/condor-xxxxxxx/local.xxxxxx/spool
```

You can know test it with the client code provided !

The web service is composed of 3 functions:

- OMElement sendJobToAved (String path): this function takes for parameter the path of the video to be processed and return an XML file with node1=the cluster number and node2=the job number
- OMElement getJobStatus (int cluster, int job): this function takes the cluster number and the job number and return the job status (1: Idle, 2:Running, 3:Removed, 4:Completed, 5:Held).
- OMElement getResult (int cluster, int job): this function takes the cluster number and the job number and return the XML result of the processing.

The web service client is basicly a java code what send a job to AVED wait until the status id is at 4 (completed) and get the result.

## Installing the AVED web-service as a service on Linux

- Create tomcat.sh file in /etc/profile.d then add the following to it:

```
export JAVA_HOME=<full-path -to-jdk>
export MBARIVISIONROOT=<full-path-to-mbarivision>
export CONDOR_SPOOL=/opt/condor-xxxxxxx/local.xxxxxx/spool
```

- Create tomcat file in /etc/init.d then add the following to it:

```
. /etc/profile.d/tomcat.sh
<full-path-to-tomcat>/bin/catalina.sh start
```

\* Add soft links to the tomcat startup script

```
ln -s /etc/init.d/tomcat /etc/rc2.d/S100tomcat
ln -s /etc/init.d/tomcat /etc/rc2.d/K100tomcat
```

Start the tomcat service

```
service tomcat start
```

## AVEDac Install Howto

---

This page last changed on Mar 31, 2010 by [dcline](#).

**Danelle E. Cline**

Copyright 2008 MBARI

### **Abstract**

This document is the complete installation guide for the AVED software.

### **Table of Contents**

[Introduction to AVEDac](#)  
[Installing AVEDac](#)

## AVEDac Installation

This page last changed on Aug 30, 2010 by [dcline](#).

# How do I install AVEDac ?

AVEDac is distributed as source files that you must compile yourself. Sorry - there are currently no binaries available. We also distribute our scripts to make it easier for you to process your video. Our development platform is Linux and these instructions have been tested on Fedora Core 3, 6, 8, 9 and 10. Fedora 11 is not supported. We have not tested these under Windows with Cygwin or on Mac OS X, but if you would like to try, please let us know what additional libraries and packages you needed to get this to work.

Installing the AVEDac software is not for the novice Linux user and requires understanding of how to install RPMs, compile code, and run Bash and Perl scripts on the Linux operating system.

There are three main components in the software suite:

| AVEDac module name | function                                                                                                                                                                                                  |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| aved-mbarivision   | Detection and tracking software                                                                                                                                                                           |
| aved-pmbarivision  | Parallel version of mbarivision designed to run on a <a href="#">Beowulf cluster</a> . The general installation instructions are the same; Beowulf specific configuration is noted <a href="#">here</a> . |
| aved-classifier    | Classification software                                                                                                                                                                                   |
| aved-ui            | Graphical user interface software used to edit results and run the classification software                                                                                                                |

### Detailed Installation Steps (aved-mbarivision/aved-pmbarivision)

- [Step 1. Build iLab Saliency Toolkit](#)
- [Step 2. Build and Install XML Libraries](#)
- [Step 3. Build and Install Transcode Software](#)
- [Step 4. Build and Install Quicktime or OpenQuicktime \(optional\)](#)
- [Step 5. Build and install Berkeley MPEG Encoder \(optional, but required for using our scripts\)](#)
- [Step 6. Build and Install mbarivision/pmbarivision](#)

### Detailed Installation Steps (aved-ui and aved-classifier) TBD

## AVED GUI Installation Notes ( Mac OS X )

---

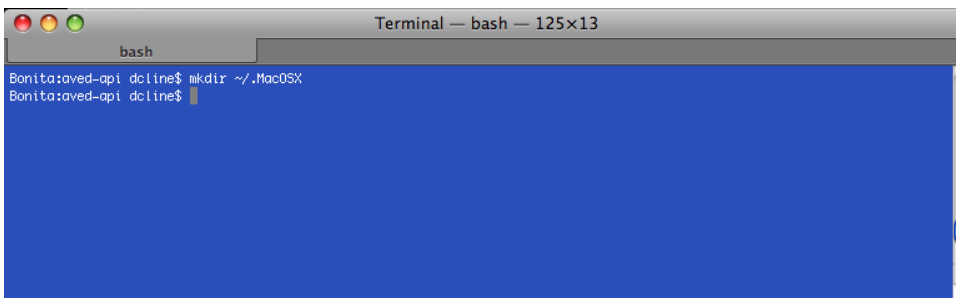
This page last changed on Aug 17, 2009 by [dcline](#).

# Mac OS X 10.5 Installation

The AVED GUI requires access to the same AVED binary and scripts that are used in the command-line. These directories are defined by environment variables.

The environment file searched each time a user logs in on a Mac is: `~/MacOSX/environment.plist` (be careful - this is case sensitive). This file is simply a property list of keys and values that the login system will read and load into the process environment of all applications that are launched when the user logs in. These variables are the same as environment variables that can be created in a command line shell (eg: `sh` or `csh`), but they also can be seen by GUI applications. These environment variables are somewhat similar to Windows' Environment User Variables. Three basic steps are required to define these 1) create the `~/MacOSX` directory, 2) add the environment variables, and 3) **Logout and Log back in to impose the changes.**

1. If you don't have a `~/MacOSX` directory, you will need to create one. Do this in a terminal window, or whatever you choose, e.g.

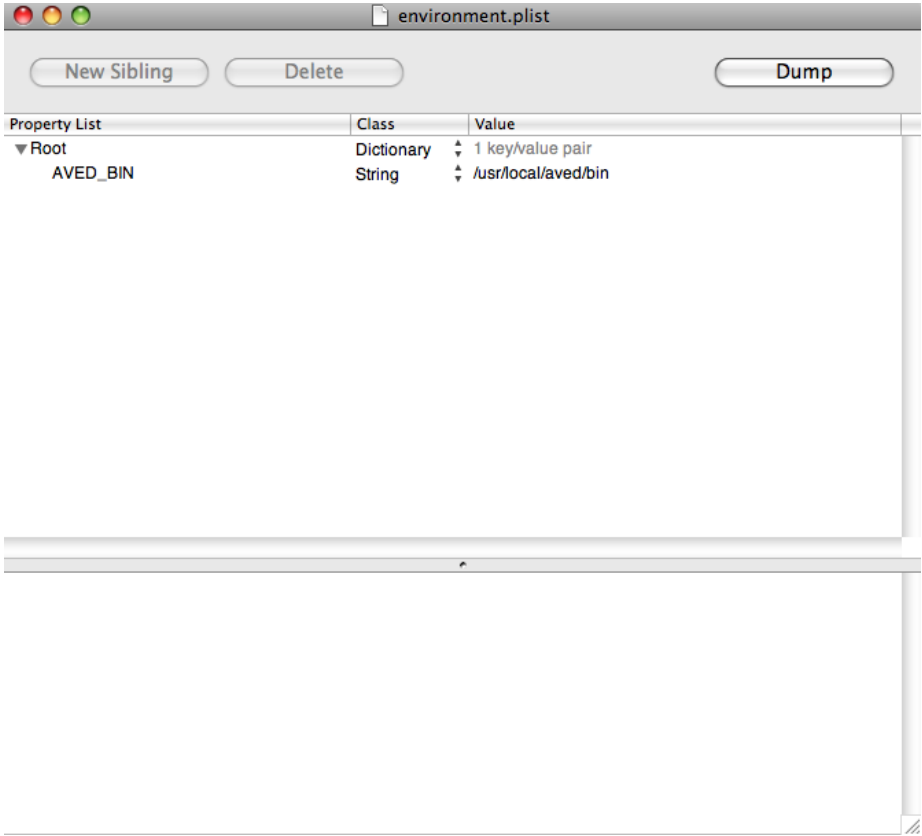


1. Next, set these variables using the `/Developer/Applications/PropertyListEditor.app` tool. For the default AVED installation, add the following as a child to the root installation to define the location of the AVED binaries and scripts (this assumes a global installation in the `/usr/local/aved/` directory):

```
AVED_BIN=/usr/local/aved/bin
AVED_SCRIPTS=/usr/local/aved/scripts
```

Then add these to the PATH environment variable using a PATH key and value. **If you want your AVED install in a non-standard location change these variable to the correct location of**

**your installation.** Your environment list in the property editor should look like the following when



correctly done:

## AVEDac Beowulf Cluster Installation Notes

---

This page last changed on Jun 09, 2010 by [dcline](#).

# Configuration

Your Beowulf cluster should have a NFS shared /home directory which is typically the way your cluster is configured it already configured. You will need at least 7 nodes to run the parallel version of the detection and tracking software called pmbarivision. There are also specific firewall and network settings required to run the pmbarivision through MBARI scripts noted in the instructions below.

## Install Cluster Command Tool (optional)

If your cluster is configured with the Using the Open Source Cluster Application Resources (OSCAR), you already have this tool and you may skip this step.

As the *root* user, download and install Cluster Command Tool version 3.1 <http://www.csm.ornl.gov/torc/C3/> (this also must be installed on all nodes in the cluster. see above site for more information). If nodes are removed or added to the cluster, you must update this list and restart the mpd service. This utility must be installed first to give you the ability to push files and execute commands easily on your worker nodes.

1. Download and install to /opt/c3-3.
2. Add a file /etc/c3.conf, e.g. for an 8-node cluster with master node named *beowulffish*

### **/etc/c3.conf**

```
cluster oscar_cluster {
  beowulffish
  dead remove_line_for_0-indexing
  node1.private.net
  node2.private.net
  node3.private.net
  node4.private.net
  node5.private.net
  node6.private.net
  node7.private.net
  node8.private.net
}
```

## Install MPI libraries

The Message Passing MPI libraries are required to build pmbarivision. If you already have the MPI development libraries installed, then all that is required is to configure your mpd.hosts file (step 2).

1. As *root* user, download and install [MPI](#) libraries to /opt/mpich-2.1.1p1

```
tar -zxvf mpich2-1.1.1p1.tar.gz
cd mpich2-1.1.1p1
./configure --prefix=/opt/mpich-2.1.1p1
make
cd src/mpe2; make
cd ..
make install
```

2. Create and add the following to /opt/mpich2-1.1p1/etc/mpd.hosts

```
touch /opt/mpich2-1.1p1/etc/mpd.hosts
chmod a+r /opt/mpich2-1.1p1/etc/mpd.hosts
```

### **/opt/mpich2-1.1p1/etc/mpd.hosts**

```
master.private.net
node1.private.net
node2.private.net
```

```
node3.private.net
node4.private.net
node5.private.net
node6.private.net
node7.private.net
node8.private.net
```

3. Push the libraries to the other nodes:

```
cpush /opt/mpich-2.1.1p1
```

## Add user *aved*

It is recommended to install and run the pmbarivision binary as a user *aved* that is visible across all nodes. To support this, create a user *aved* for installing and running pmbarivision. As the *root* user:

```
groupadd -g 300 aved
useradd -c "AVED" -g aved -u 300 aved
```

The *aved* user and group id must be synchronized across all nodes and machines in the Beowulf cluster. Once the *aved* user is created, syncing up the user is typically done automatically by the OSCAR Password Installer and User Management (OPIUM) software, but you can also do this manually with:

```
/opt/opium/bin/sync_users -f
```

## Setup the user profile

Add to *aved.sh/csh* in the */etc/profile.d/* your MPDIR and SCRATCH\_DIR settings. MPDIR points to the root of your mpich installation. SCRATCH\_DIR is an NFS mounted directory that is shared between nodes where video will be processed. SCRATCH\_DIR can be a separate disk, a separate partition, or a NFS shared directory - it just represents a temporary "scratch" location that is shared across nodes where video will be temporarily placed for processing. SCRATCH\_DIR must have read and write access for all users.

### */etc/profile.d/aved.sh*

```
export MPDIR=/opt/mpich2-1.1.1p1
export PATH=$MPDIR/bin:$PATH
export SCRATCH_DIR=/mnt/scratch
```

### */etc/profile.d/aved.csh*

```
set MPDIR = /opt/mpich2-1.1p1
set PATH = ($MPDIR/bin $PATH )
set SCRATCH_DIR = /mnt/scratch
```

When done push these to the other nodes:

```
cpush /etc/profile.d/aved.* /etc/profile.d/
```

## Test password-less SSH across all nodes

Password-less ssh login is required to run the AVED MPI jobs. Your machine may already be configured for this. To test this, switch to the user *aved* and try ssh into the first node1. You should **not** be prompted for a password

```
root@beowulffish ~]# su aved
[aved@beowulffish ~]$ ssh n01
Last login: Thu Apr 15 10:54:27 2010 from master.private.net
[aved@node1 ~]$
```

If your machine is not configured for password-less login, there is a wealth of information on how to set this up on the web. However, this is typically configured by default in the OSCAR tool suite in */etc/profile.d/ssh-oscar.ssh/csh*.

## Copy ffmpeg library dependencies to all nodes

These dependencies are required in the pmbarivision code. This is the only dependency needed outside those installed in the /home/aved NFS shared directory. Push dependencies to the client nodes and refresh the dynamic linker:

```
cpush /usr/lib/libavcodec.so.51
cpush /usr/lib/libavcodec.so.52
cpush /usr/lib/libavcodec.so.51.40.4
cpush /usr/lib/libavformat.so.51.12.1
cpush /usr/lib/libavformat.so.51
cpush /usr/lib/libavformat.so.51.12.1
cpush /usr/lib/libavutil.so
cpush /usr/lib/libavutil.so.49
cpush /usr/lib/libavutil.so.49.4.0
cpush /usr/lib/libSDL-1.2.so.0
cpush /usr/lib/libSDL-1.2.so.0.7.0
cexec 'ldconfig'
```



## AVEDac Beowulf Cluster Miscellaneous Settings

---

This page last changed on Apr 21, 2010 by [dcline](#).

### Pam.d settings

As *root* user, in */etc/pam.d*, check your settings against these. These are correct to allow for *rsh*, *ssh*, and *rexec* across nodes, but not *rlogin*.

#### **/etc/pam.d/rexec**

```
#auth      required    pam_nologin.so
#auth      required    pam_securetty.so
```

#### **/etc/pam.d/rsh**

```
auth      sufficient    pam_nologin.so
auth      optional      pam_securetty.so
auth      sufficient     pam_env.so
auth      sufficient     pam_rhosts_auth.so
account    sufficient     pam_stack.so service=system-auth
session    sufficient     pam_stack.so service=system-auth
```

#### **/etc/pam.d/rlogin**

```
auth      sufficient     pam_rhosts_auth.so
#auth      required      pam_securetty.so
auth      required       pam_nologin.so
auth      required       pam_env.so
auth      required       pam_stack.so service=system-auth
account    required       pam_stack.so service=system-auth
password   required       pam_stack.so service=system-auth
session    required       pam_stack.so service=system-auth
```

#### **/etc/pam.d/sshd**

```
auth      required       pam_stack.so service=system-auth
auth      sufficient     pam_nologin.so
account    required       pam_stack.so service=system-auth
password   required       pam_stack.so service=system-auth
session    required       pam_stack.so service=system-auth
session    required       pam_limits.so
session    optional      pam_console.so
```

### SSH settings

Check if the following are in */etc/ssh/ssh\_config*,

#### **/etc/ssh/ssh\_config**

```
Host *
ForwardX11 yes
StrictHostKeyChecking no
UsePrivilegedPort no
```

then restart the service

```
service sshd restart
```

## AVEDac Beowulf Cluster MPICH Testing Notes

---

This page last changed on Apr 20, 2010 by [dcline](#).

### Install mpd service

This is used to configure a mpd ring running under the root account that will be used by users. This assumes you have 4 CPUs per each node and you may need to adjust this according to your cluster.

1. Create mpd service file similar to the following  
**/etc/rc.d/init.d/mpd**

```
#!/bin/sh
#
# Start or stop system wide mpd daemon

# Source the aved functions
. /etc/profile.d/aved.sh

case "$1" in
start)
# This assumes only nodes and not the master are listed in the mpd.hosts file
# because the master is not typically listed as a worker node
c=`wc -l $MPDIR/etc/mpd.hosts | awk '{print $1}'`
# Add one to include the master
((c += 1))
if [ ! -e '/tmp/mpd2.console_root' ]; then
mpdboot --ncpus=4 --totalnum=$c --file="$MPDIR/etc/mpd.hosts --mpd=$MPDIR/bin/mpd.py"
else
echo "restarting mpd"
mpdallexit
mpdboot --ncpus=4 --totalnum=$c --file="$MPDIR/etc/mpd.hosts --mpd=$MPDIR/bin/mpd.py"
fi
mpdtrace
;;

stop)
if [ -e '/tmp/mpd2.console_root' ]; then
mpdallexit
#remove tmp file in case mpdallexit cmd fails
if [ -e '/tmp/mpd2.console_root' ]; then
rm /tmp/mpd2.console_root
fi
echo "mpd daemon stopped"
else
echo "mpd daemon already stopped"
fi
;;

status)
if [ -e '/tmp/mpd2.console_root' ]; then
mpdtrace -l
else
echo "mpd daemon not running"
fi
;;

*) echo "usage: $0 {start|stop|status}"
;;
"/etc/rc.d/init.d/mpd" 53L, 1411C
```

2. Then push to the other nodes

```
cpush /etc/rc.d/init.d/mpd /etc/rc.d/init.d/mpd
```

3. Start the mpd service

```
service mpd start
```

You should see the nodes all come up, something like (not necessarily in this order):

```
beowulffish
node1
node4
node3
node2
node5
node8
node6
node7
```

## mbarivision Installation - Step 5. Build and install Berkeley MPEG Encoder (optional)

---

This page last changed on Jun 09, 2010 by [dcline](#).

The Berkeley mpeg encoder is used in the AVED scripts to create mpeg clips of the results from mbarivision. If you don't need to create video clips of the output, or have some alternative tool you use, then skip this step.

### Install GraphicsMagick

GraphicsMagick is required at various points in the scripts. It's also a very handy tool in general for image handling. If you don't already have this installed, install GraphicsMagick with:

```
yum install GraphicsMagick
```

### Build and install Berkeley mpeg encoder

Download the Berkeley mpeg encoder from: [http://bmrc.berkeley.edu/frame/research/mpeg/mpeg\\_encode.html](http://bmrc.berkeley.edu/frame/research/mpeg/mpeg_encode.html) If this site is down, we have copy here: [mpeg\\_encode.tgz](#).

We ran into difficulties compiling this, and needed to make a minor modifications to libpnmrw.c/h files. Replace the libpnmrw.c/h files in your download with these: [libpnmrw.c](#) [libpnmrw.h](#).

Build and install this with:

```
cp </my/downloads/libpnmrw.c> <mpeg_encode/libpnmrw.c>
cp </my/downloads/libpnmrw.h> <mpeg_encode/headers/libpnmrw.h>
make
install mpeg_encode /usr/local/bin
```

If installing on a Beowulf cluster, install this to a NFS directory shared across nodes

```
install mpeg_encode /home/shared/aved/bin
```

When you are done, complete installation with [Step 6. Build and Install Mbarivision or Pmbarivision](#).

## mbarivision Installation - Step 1. Build iLab Saliency Toolkit

This page last changed on Jun 02, 2010 by [dcline](#).

# Getting the Saliency Toolkit

Most importantly, mbarivision requires the Saliency Toolkit from the [ilab](#) at USC. The Saliency toolkit is distributed as source files, so you must compile it. To get the Toolkit, you must ask for permission to use it by following the instructions on the bottom of this page: <http://ilab.usc.edu/toolkit/downloads.shtml>. Let them know you will be using this with the AVED software from MBARI. As long as you are using the Toolkit for academic or research use, you will be granted access to the code. You will be emailed a user login and password to check out the code from the ilab SVN repository. If you don't have [Subversion](#) installed, you can install it using:

```
yum install subversion
```

Once you get the password to the iLab repository, check-out the saliency code from the date 2008-04-15 using svn. **Important** - we unfortunately are working with an older version of the toolkit because we haven't had time to port our code to work with the latest saliency revision. So, to be compatible with the version we are working with, you must **checkout the release on the date 2008-04-15**. The checkout command:

```
svn checkout --username anonsvn svn:///ilab.usc.edu/trunk/saliency --revision {2008-04-15} saliency
```

# Requirements for building the Saliency Toolkit

Next, install the Saliency Toolkit library dependencies using yum if needed. These are the dependencies we have found from our last experience installing it on Fedora Core 9 that may be needed. These may already be installed on your system, but it's always a good idea to check first:

| Library                            | RPM Packages                                                | Fedora command                       |
|------------------------------------|-------------------------------------------------------------|--------------------------------------|
| libX11                             | xorg-x11-proto-devel                                        | yum -y install xorg-x11-proto-devel  |
| gcc                                | gcc-c++                                                     | yum -y install gcc-c++               |
| tclsh                              | tclx                                                        | yum -y install tclx                  |
| libXShm-devel                      | libXext-devel                                               | yum -y install libXext-devel         |
| libbz2-devel                       | bzip2-devel                                                 | yum -y install bzip2-devel           |
| libz-devel                         | zlib-devel                                                  | yum -y install zlib-devel            |
| libxml2-devel                      | xml2-devel                                                  | yum -y install xml2-devel            |
| lpng                               | libpng and libpng-devel                                     | yum -y install libpng libpng-devel   |
| ljpeg                              | libjpeg and libjpeg-devel                                   | yum -y install libjpeg libjpeg-devel |
| libavcodec, libavutil, libavformat | ffmpeg and ffmpeg-devel or install from sources (see below) |                                      |

You will also need the gcc compiler version gcc-3.4.X. We suggest 3.4.3. If you are running a more recent version e.g. gcc-4.3 it will not compile correctly.

Download gcc from [here](#):

<http://ftp.gnu.org/gnu/gcc/gcc-3.4.3/>

This is a good site to orient you on how to install a second gcc compiler:

[http://www.faqs.org/docs/ldev/0130091154\\_71.htm](http://www.faqs.org/docs/ldev/0130091154_71.htm)

When the new compiler is installed, you can use this new compiler in the saliency build by setting the following environment variables to point to your gcc-3.4.3 version before running configure.

| Environment Variable | Description                                                                                                             |
|----------------------|-------------------------------------------------------------------------------------------------------------------------|
| CC                   | C compiler command                                                                                                      |
| LDFLAGS              | linker flags, e.g. -L<lib dir> if you have libraries in a nonstandard directory <lib dir>                               |
| CPPFLAGS             | C/C++/Objective C preprocessor flags, e.g. -I<include dir> if you have headers in a nonstandard directory <include dir> |
| CXX                  | C++ compiler command                                                                                                    |

## Building ffmpeg from source

If you cannot get ffmpeg in a RPM, to install ffmpeg from sources, get the last version. Checkout ffmpeg, build, and install (this assumes you have sudo permissions to install applications to /usr):

```
svn checkout svn://svn.mplayerhq.hu/ffmpeg/trunk ffmpeg
cd <path-to-ffmpeg>
./configure --enable-shared --prefix=/usr
make
sudo make install
```

## Building the Saliency Toolkit

When all Saliency dependencies are installed, the required build command to work with the AVED software, is the make *core* and *tprogs* build rules with the following command:

```
cd <path-to-saliency>
./configure --without-qtdir --enable-force32
make core
make tprogs
```



### Warning

Don't change these configuration options or it will break the AVED mbarivision build. You can add options like --prefix, --enable-quietcompile, etc. but don't remove the options above. If you want to experiment with the saliency toolkit, copy it to a separate directory that won't be used in the AVED mbarivision build !

Now, go get a cup of coffee, because this step takes a while. It took 28 minutes on a dual-XEON 2.4 GHz server. When you are done, go to [Step 2. Build and Install XML Libraries](#)

## mbarivision Installation - Step 2. Build and Install XML Libraries

---

This page last changed on Jun 09, 2010 by [dcline](#).

### Build and install the Xerces C++ library version 2.7

Xerces C++ is a library allowing AVED to export the results into an XML file. This library provides several utilities to manipulate and to validate XML documents using the DOM and SAX APIs. Check out the code from apache, and install with:

#### For mbarivision build, install to /usr/local

```
svn co https://svn.apache.org/repos/asf/xerces/c/tags/Xerces-C_2_7_0
export XERCESCROOT=`pwd`/Xerces-C_2_7_0
cd $XERCESCROOT/src/xercesc
./runConfigure -p linux -cgcc -xg++ -minmem -nsocket -tnative -rpthread -P /usr/local
gmake
gmake install
```

#### For pmbarivision build, install to your shared directory across nodes, e.g. /home/shared/aved:

```
svn co https://svn.apache.org/repos/asf/xerces/c/tags/Xerces-C_2_7_0
export XERCESCROOT=`pwd`/Xerces-C_2_7_0
cd $XERCESCROOT/src/xercesc
./runConfigure -p linux -cgcc -xg++ -minmem -nsocket -tnative -rpthread -P /home/shared/aved
gmake
gmake install
```

And update your dynamic loader across all nodes to find this library

```
echo /home/shared/aved/lib >> /etc/ld.so.conf
cpush /etc/ld.so.conf
cexec ldconfig -v
```

### Install the simple XML writer

Install CPAN and add XML perl components for writing simple XML files used in the AVED scripts. If you don't plan on using the AVED scripts, you can skip this step.

This requires the perl-CPAN installer, if you don't have perl-CPAN installed, install it with:

```
yum install perl-CPAN
```

then, install the Simple and Writer modules with:

```
perl -MCPAN -e 'install XML::Simple'
perl -MCPAN -e 'install XML::Writer'
```

When you are done, you can skip to [step 6](#), or continue with the next optional step [AVED Installation Step 3. Build and Install Transcode Software](#).

## mbarivision Installation - Step 3. Build and Install Transcode

This page last changed on Jun 22, 2010 by [dcline](#).

### About the Transcode software

Transcode is a suite of command line utilities for transcoding video and can be used to convert clips into individual PPM frames. In the provided scripts it is used to convert frames to clips and for basic image handling operations.

There are other free tools available like mencoder, but we have found the transcode software to be the most flexible and support the widest range of formats.

### Transcode RPM installation

Transcode can be installed from a RPM using the yum command, but it requires RPMS from the [livna](#) and [freshrpms](#) repositories so you'll need to add support for both of these repositories first, before installing it with the yum command.

1. We had the best luck with the livna repository, so follow the instructions here to add the livna repository: [livna](#)
2. Download and install the correct Fedora core version of freshrpm, e.g. for FC3 download and install: <http://ftp.freshrpms.net/pub/freshrpms/fedora/linux/3/freshrpms-release/>
3. Update the yum repository, and install transcode using the yum command:

```
yum update
yum install transcode
```

### Transcode source code installation (not recommended)

If you cannot find a transcode RPM, you can get the source code from <http://www.transcoding.org/cgi-bin/transcode> and install it, **however this is not recommended**. There are many dependencies to transcode and the build can easily break. Don't do this step unless you absolutely have to.

First download, then uncompress.

```
tar jxvf transcode-xxxxxxx.tar.bz2
```

Transcode uses many shared libraries, and you may need to download and install them, depending on what you have installed on your system. Here is the list of some of the RPM packages that we have found are needed:

| Library  | RPM Packages                | Fedora command                         |
|----------|-----------------------------|----------------------------------------|
| mpeg2dec | mpeg2dec and mpeg2dec-devel | yum install mpeg2dec<br>mpeg2dec-devel |
| lvo      | lvo and lvo-devel           | yum install lvo lvo-devel              |
| libXv    | libXv-devel                 | yum install libXv-devel                |

Next, go in the transcode directory and build it. This example disables lame and libdvdrread and enables libquicktime which worked on our system. Your system may be some variant of this:

```
cd transcode-xxxxxxx
./configure --disable-lame --disable-libdvdrread --enable-libquicktime --enable-imagemagick
make
sudo make install
```

When you are done, skip to [step 6](#), or continue to the next optional step [AVED Installation - Step 4. Build and Install Quicktime or OpenQuicktime](#).



## Mac OS X installation

First, if you don't have yum, install it on your mac. A version that worked on Mac OS X 10.5 is here: <http://transcode.darwinports.com>. Download the latest version and follow the above instructions for Linux.

When you are done you can skip to [step 6](#), or continue to the next optional step [AVED Installation - Step 4. Build and Install Quicktime or OpenQuicktime](#).

## mbarivision Installation - Step 4. Build and Install Quicktime or OpenQuicktime (optional)

---

This page last changed on May 12, 2010 by [dcline](#).

# About libquicktime

If you have video in a Quicktime container (.mov file) you may need a quicktime library to decode the video. This is used in conjunction with the transcoding software described in the next step - [mbarivision Installation - Step 3. Build and Install Transcode](#). If you are not sure, install this anyway. It doesn't hurt to have this on your system.

# Installing libquicktime

Installation can be done from your installer, or from source code. Here are a couple of options:

## Yum installation

As root run:

```
yum install libquicktime libquicktime-devel
```

## Source code installation

Download source from <http://libquicktime.sourceforge.net/>.



### Warning

Libquicktime has many dependencies so your installation may require installing other dependent libraries. See <http://libquicktime.sourceforge.net/> for more detailed instructions, otherwise you can try a simple install like:

```
cd <full/path/to/quicktime>
./configure
make
sudo make install
```

# About OpenQuicktime

Early in the AVED development we put our video into Quicktime containers and added a timecode track that corresponded to the time track recorded on our tapes. The SMPTE 12M standard is used in video and in film for specifying time. This is what comes off our tape-decks and what was added to the Quicktime timecode track. See for more information: [http://en.wikipedia.org/wiki/SMPTE\\_time\\_code](http://en.wikipedia.org/wiki/SMPTE_time_code). We did this by modifying the Linux OpenQuicktime library to support reading and writing a single timecode track.

We no longer support OpenQuicktime, but include it here for backwards support of our older digitized video. We now maintain the time code throughout our processing by naming the video according to the [ISO8601](#) standard, and then maintain the timecode track in the xml formatted results of the output. This allows us more flexibility in supporting different video containers, and since support for OpenQuicktime is diminishing, abandoning it was a sensible step.

## Building and Installing OpenQuicktime

The custom version of OpenQuicktime that includes the ability to write and read a single timecode track can be found on our internal CVS server Moonjelly. CVS check out the `aved/OpenQuicktime` module, build, then install. The following instructions assume you have access to, and know how to check-out code from the internal MBARI CVS server Moonjelly. If you don't, then see these instructions for more information: [welcome to CVS](#).

First check-out the module:

```
cvcs co aved/OpenQuicktime
```

Then build according to the following:

```
cd <full/path/to/OpenQuicktime>
./configure
make
sudo make install
```

When you are done, you can skip to [step 6](#), or continue to the next optional step [AVED Installation - Step 5. Build and install Berkeley MPEG Encoder](#)

## mbarivision Installation - Step 6. Build and Install mbarivision or pmbarivision

This page last changed on May 26, 2010 by [dcline](#).

# About mbarivision and pmbarivision

Mbarivision is the name of the main executable in the AVED software. Mbarivision got its name because it is built upon the Saliency Toolkit and supports similar commands to the **ezvision** binary also build with Toolkit, thus the name mbarivision.

Pmbarivision is a parallel version of the Mbarivision software. Pmbarivision is a rewritten version of mbarivision, designed to achieve speed-up through a combination of pipelining the mbarivision algorithm and distributing saliency algorithm computation across nodes in a Beowulf cluster. It uses the [MPICH2](#) message passing libraries. Pmbarivision is used together with a second executable pvisionTCPmbari

Both are command line programs with options available using the --help switch, e.g.

```
mbarivision --help
```

A more detailed description of the options in mbarivision can be found [here](#).

## Getting (p)mbarivision

Mbarivision has always been an experimental component of the AVED software. It is the main component in the AVED software suite that performs the automated detection and tracking. Classification is done in the aved-classification module. To get the latest version:

1. Install Mercurial (<http://mercurial.selenic.com/wiki/>). On Linux, run as root user:  

```
yum install '*mercurial*'
```
2. Clone the repository to your local machine <http://code.google.com/p/avedac/source/checkout>

## Building and Installing (p)mbarivision

There are two methods to install: either through a Makefile, or through a Maven build. The Maven build is meant to be used when installing the entire code suite. If building pmbarivision, you **must build mbarivision first because it's a dependency in the pmbarivision build**.

### Mbarivision Makefile install

1. Change into the source directory:  

```
cd /full/path/to/avedac/aved-mbarivision/src/main/cpp
```
2. Run configure with the saliency and xercesc paths pointing to the installed [saliency](#) and [xerces](#) code of your installation, e.g.  

```
./configure --prefix=${HOME}/aved --with-saliency=${HOME}/saliency --with-xercesc=${HOME}/Xerces-C_2_7_0
```
3. Now install. The executables will be installed in the subdirectory bin defined by the --prefix option, e.g. in the above example \${HOME}/aved/bin. Make sure you have the appropriate permissions to write to the directory.  

```
make all install
```

### Pmbarivision Makefile install

1. **Build mbarivision first because it's a dependency in the pmbarivision build.**
2. Test whether the path to the mpicxx compiler is visible in your PATH, if not then add it, e.g.  

```
export PATH=$PATH:/opt/mpich2-1.1p1/bin
```
3. Change into the source directory:

```
cd /full/path/to/avedac/aved-pbarivision/src/main/cpp
```

4. Run configure with the saliency and xercesc paths pointing to the installed [saliency](#), [xerces](#), and the aved-mbarivision dependency of your installation. e.g.

```
./configure --prefix=${HOME}/aved --with-saliency=${HOME}/saliency --with-xercesc=${HOME}/Xerces-C_2_7_0 --with-mbarivision=${HOME}/avedac/aved-mbarivision
```



#### Handy Hint

If you update either the mbarivision or Saliency source code, be sure to run a **make allclean**. This will clean not only the object files, but the a generated dependencies file that is used to compile each source file.

1. Now install. The executables will be installed in the subdirectory bin defined by the --prefix option, e.g. in the above example \${HOME}/aved/bin. Make sure you have the appropriate permissions to write to the directory.

```
make all install
```



#### Important

The install --prefix must be in a shared NFS directory for the pmbarivision code to run correctly.

### Mbarivision Maven Install

1. Install Java  
Install Java from TBD and set your JAVA\_HOME environment variable
2. Install Maven  
Maven is based on the concept of a project object model (POM). After you download the code, if you look in each of the major subdirectories in this software you will see a pom.xml file, and in each of these files is a description of the build and its dependencies, whether it be a java build, c++ build, or otherwise.  
Download Maven from <http://maven.apache.org/download.html>
3. Modify the pom.xml  
Edit the pom.xml in the root directory of the checked-out code to match your installation. These paths point to the installed [saliency](#) and [xerces](#) code of your installation.

#### /full/path/to/avedac/pom.xml

```
<profiles>
<properties>
...
<installPath>${HOME}/aved</installPath>
<xercesRoot>${HOME}/Xerces-2_7_0</xercesRoot>
<saliencyRoot>${HOME}/saliency</saliencyRoot>
</properties>
</profiles>
```

To build and install the mbarivision executable:

```
mvn reactor:make -Dmake.folders=aved-mbarivision
```

To build and install the pmbarivision and pvisionTCPmbari executables:

```
mvn reactor:make -Dmake.folders=aved-pmbarivision
```

The binaries will be installed to the directory defined by the property <installPath> in the parent pom.xml. Make sure you have the appropriate permissions to write to the directory.

## AVEDac Introduction

---

This page last changed on Mar 30, 2010 by [dcline](#).

### Overview of the AVEDac project

In order to study the distribution and abundance of oceanic animals, MBARI and other oceanographic institutes, use high-resolution video equipment on remotely operated vehicles. Quantitative video transects (QVTs) supplant traditional net tows to assess the quantity and diversity of organisms in the water column. QVTs are run from 50 m to 4000 m and provide high-resolution data at the scale of the individual animals as well as their natural aggregation patterns. However, the current, manual method of analyzing QVTs is labor intensive and tedious.

AVEDac is a software solution designed for automating the detection of animals in underwater video to enhance the productivity of human video annotators. It was developed by the Monterey Bay Aquarium Research Institute in collaboration with the [University of Southern California](#) and the [California Institute of Technology](#). AVED is based on the USC [iLab Neomorphic Visual C++ Toolkit](#) and is designed to detect, track, and classify animals in underwater.

The continued use of ROVs and future use of Autonomous Underwater Vehicles (AUVs) for QVTs offer potential for even more data, perhaps many times what we current collect and analyze. Hence, we see tremendous benefit in automating portions of the analysis. We also see great benefit in automating analysis of video from fixed ocean observatory cameras, where autonomous response to potential events (pan/zoom to events), and automated processing of largely "boring" (event sparse) video streams from 10s or 100s or even 1000s of network cameras could be key to those cameras being useful practical scientific instruments.

### User Support

See the [AVEDac Support](#) for help.

## AVEDac Support

---

This page last changed on Jun 02, 2010 by [dcline](#).

### FAQ

Start by reading the [FAQ](#) page before contacting the developer. There is currently only one developer supporting AVED and she is very busy, but happy to help if she can. Sorry, but there is no user forum or mailing list yet.

### Reporting bugs and asking questions

Send your question in an email to Danelle Cline ([dcline@mbari.org](mailto:dcline@mbari.org)). Please include the version you are using and any system details that are relevant to your problem. In mbarivision/pmbarivision the version can be seen with the --version flag. In the graphical interface, the version is in the "About" box.

## AVEDac FAQ

---

This page last changed on Jun 09, 2010 by [dcline](#).

### runclip Xercesc error

#### Symptom

Can't locate XML/Xerces.pm in @INC (@INC contains: /usr/lib64/perl5/5.10.0/x86\_64-linux-thread-multi /usr/lib/perl5/5.10.0 /usr/local/lib64/perl5/site\_perl/5.10.0/x86\_64-linux-thread-multi /usr/local/lib/perl5/site\_perl/5.10.0 /usr/lib64/perl5/vendor\_perl/5.10.0/x86\_64-linux-thread-multi /usr/lib/perl5/vendor\_perl/5.10.0 /usr/lib/perl5/vendor\_perl .) at /usr/local/aved/scripts/extract\_source\_metadata.pl line 62.  
BEGIN failed--compilation aborted at /usr/local/aved/scripts/extract\_source\_metadata.pl line 62.

#### Solution

### clip2ppm script segmentation fault

#### Symptom

/usr/local/aved/scripts/clip2ppm: line 246: 19759 Segmentation fault      \$cmd

#### Solution

Appears to be related to selinux. It's not recommended, but you can disable this. See[ [this page|http://www.crypt.gen.nz/selinux/disable\\_selinux.html#DIS1](http://www.crypt.gen.nz/selinux/disable_selinux.html#DIS1)]for help. Otherwise, the TRAP will catch this and return



# Runing mbarivision

mbarivision is invoked from command line and has many many command line options. A list of the available command options can be found [here](#).

## Examples

In mbarivision/samples a movie file composed of several frame is provided. The command line examples are executed from the mbarivision directory.

| Goal                                                      | Command                                                                                                     | Outputs                                     | Comments                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-----------------------------------------------------------|-------------------------------------------------------------------------------------------------------------|---------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Detect events and outline the events in the output frames | <code>mbarivision --in=raster:samples/f#.ppm --out=display --input-frames=0-99@1 --mbari-save-output</code> | frames with events outlined in bounding box | For a full list of options in the runscript script, simply run the script with no arguments. There are several options here. The --in option says to take input from the named raster file; several raster file formats are supported, including png and pnm (ppm/pgm/pbm). ]. When you pass the filename to --in, put a '#' in the place where the frame number should go. Thus, it will read f000000.ppm, f000001.ppm. <b>It is important to format the filenames with a single dot '.' e.g. a filename formatted like T.f00000.ppm isn't supported, but Tf00000.ppm is okay, in this case you would pass the in argument with Tf#.ppm.</b> By default, it will keep reading input files until it encounters a missing raster file. If you want it to stop sooner, you can specify a frame range with the --input-frames option, such as --input-frames=0-99@1. Same than the previous example, but this time the output will be a sequence of 100 |

|                                                                                                                         |                                                                                                                                                                                                                                  |          |                                                                                                                                                                                                                    |
|-------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                                                         |                                                                                                                                                                                                                                  |          | frames with the events detected outlined by a bounding box.                                                                                                                                                        |
| Detect events, but change the tracking mode, cache size, and min/max event areas, and save the result into an XML file. | mbarivision --in=raster:samples/f#.ppm --out=display --input-frames=0-99@1 --mbari-cache-size=15 --mbari-tracking-mode=BoundingBox --mbari-min-event-area=100 --mbari-max-event-area=10000 --mbari-save-events-xml=./benthic.xml | XML file | Similar to above, except here we change the default tracking mode, and only keep objects with area between 100-1000 square pixels. Also the results are saved to an XML file in the samples folder as benthic.xml. |

## Running mbarivision with runclip

The main script for processing a clip with the mbarivision executable is the bash script *runclip*. *runclip* will uncompress the video into individual frames and setup the command-line arguments for running mbarivision. For a full list of the runclip options, simply run runclip with no arguments.

runclip



### Useful Information

*runclip* will not work for all video formats. It uses the transcode software. Therefore, only video that transcode can decode will work with *runclip*. The steps for installing transcode are described [here](#)

To use this script, first setup the PATH variable to include the directory *runclip* is installed in. If you have installed the scripts in, for example, your home directory in the directory *aved/scripts*

```
export PATH=~/.aved/scripts
```

### Examples

| Goal                                                                                                                               | Command                                                        | Outputs                                                                                              | Comments                                                                      |
|------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------|------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------|
| Detect events using a pre-defined set of options optimal for a benthic video, create a mpeg of the results, and apply a video mask | runclip -m videomask.jpg -f benthic -g -i 20100203T080408Z.avi | mpeg clip of the results, XML formatted metadata output suitable for editing the graphical interface | For a full list of the runclip options, simply run runclip with no arguments. |

## Running Beowulf-enabled pmbarivision

### Running pmbarivision with runclip

See the notes above for mbarivision. Setup is the same for pmbarivision.

### Examples

| Goal | Command | Outputs | Comments |
|------|---------|---------|----------|
|------|---------|---------|----------|

|                                                                                                                                    |                                                                                                     |                                                                                                      |                                                                               |
|------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------|
| Detect events using a pre-defined set of options optimal for a benthic video, create a mpeg of the results, and apply a video mask | <code>runclip -w /mnt/scratch/workers -m videomask.jpg -f benthic -g -i 20100203T080408Z.avi</code> | mpeg clip of the results, XML formatted metadata output suitable for editing the graphical interface | For a full list of the runclip options, simply run runclip with no arguments. |
|------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------|

## AVED Mbarivision Options

This page last changed on May 11, 2010 by [dcline](#).

### Mbarivision Options

The table below lists all the mbarivision options arranged in 3 categories: Input Reading/Formatting Options, Output Writing/Formatting Options and Special Features Options. There are other options, outside this table that pertain to the saliency toolkit. The entire list of options can be found using the mbarivision --help switch.

| Option                                      | Default   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|---------------------------------------------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Input Reading/Formatting</b>             |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| --input-frames=[first-last]@[delay_or_rate] | required  | Input frame range and inter-frame delay or rate. The frame range can include optional specifications for the first frame (default value=0), last frame (default=max), and/or frame step (default=1). A fixed framerate can be specified either as an inter-frame interval, with a suffix one of (s,ms,us,ns), or as a frame rate, with a suffix of Hz.                                                                                                                                                                                                                                               |
| --crop-input=x1,y1,x2,y2                    | [0,0,0,0] | Crop input frames, or 0,0,0,0 for no cropping.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| --rescale-input=<width>x<height>            | [0x0]     | Rescale input frames to fixed dims, or 0x0 for no rescaling                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| --[no]preserve-input-aspect                 | [no]      | Preserve input frame aspect ratio if rescaling to fixed dims                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| --zero-number-frames=<true false>           | [false]   | Force all input and output frames to have the number 000000                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| --in=<[type]:[spec]>                        | required  | Reads input from one or more raster files. The leading 'raster:' may be omitted if the file has one of the known raster extensions: .pnm, .pgm, .ppm, .pbm, .pfm, .p, or any of the above with an additional .gz or .bz2 extension, in which case the image file will be transparently decompressed. If the given filename includes a pair of hashes ('#'), then characters in between the hashes will be interpreted as a minimum numeric field width, and the frame number will be formatted as a zero-padded string to that minimum width, and will be substituted for the entire #nnn# sequence. |

|                                                           |               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------------------------------------------------------|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                           |               | <p>As special cases, a single '#' is equivalent to '#6#', and '##' is equivalent to '#0#'. Thus a filename of foo#.png or foo#6#.png will cause the stream to read foo000000.png, foo000001.png, etc.; foo##.png or foo#0#.png will read foo0.png, foo1.png, etc.; and foo#3#.png will generate foo000.png, foo001.png, etc.</p> <p><b>It is important to format the filenames with a single dot '.' e.g. a filename formatted like T.f00000.ppm isn't supported, but Tf00000.ppm is okay, in this case you would pass the in argument with Tf#.ppm.</b></p> |
| --mbari-load-events=fileName                              | []            | Load the event structure from a text file instead of computing it from the frames                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| --mbari-load-properties=fileName                          | []            | Load the event property vector from a text file                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|                                                           |               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>Output Writng/Formatting</b>                           |               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| --out=<raster display mpeg none>                          | required      | Value recomended is none.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| --[no]mbari-save-results                                  | [no]          | Save intermediate results in MBARI programs to disc                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| --[no]mbari-display-results                               | [no]          | Display intermediate results in MBARI programs                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| --mbari-mark-interesting=<None Shape Outline BoundingBox> | [BoundingBox] | Way to mark interesting events in output frames of MBARI programs                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| --mbari-opacity=<0.0 ... 1.0>                             | [1.0]         | Opacity of shape or outline markings of events                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| --[no]mbari-mark-candidate                                | [yes]         | Mark candidates for interesting events in output frames of MBARI programs                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| --[no]mbari-mark-prediction                               | [no]          | Mark the Kalman Filter's prediction for the location of an object in output frames of MBARI programs                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| --[no]mbari-mark-foe                                      | [no]          | Mark the focus of expansion in the output frames of MBARI programs                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| --[no]mbari-save-output                                   | [no]          | Save output frames in MBARI programs                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

|                                                 |       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-------------------------------------------------|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| --[no]mbari-display-output                      | [no]  | Display output frames in MBARI programs                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| --[no]mbari-label-events                        | [yes] | Write event labels into the output frames                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| --mbari-rescale-display=<width>x<height>        | [0x0] | Rescale displays to <width>x<height>, or 0x0 for no rescaling                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| --mbari-save-events=fileName                    | []    | Save the event structure to a text file                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| --mbari-save-properties=fileName                | []    | Save the event property vector to a text file                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| --mbari-save-positions=fileName                 | []    | Save the positions of events to a text file                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| --mbari-save-event-num=ev1,ev1,...,evN; or: all | []    | Save video clips showing specific events                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| --mbari-save-event-summary=fileName             | []    | Save a human readable summary of all the events to a text file                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| --[no]mbari-save-non-interesting-events         | [yes] | If saving events, save non interesting events in addition to interesting event. Default is to not save non interesting events.                                                                                                                                                                                                                                                                                                                                                                  |
| --mbari-save-events-xml=fileName                | []    | Save a XML output per all events                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| --mbari-source-metadata=fileName                | []    | Add video input source information to XML output                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|                                                 |       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>Special Features</b>                         |       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| -mbari-mosaic-benthic-stills                    | [no]  | Implements good choice of options to experiment with processing still images from a still or moving camera traversing the sea bottom. EQUIVALENT TO: --mbari-saliency-dist=1 --mbari-tracking-mode=None --mbari-keep-boring-WTA-points=yes --mbari-save-non-interesting-events=yes --mbari-segment-algorithm-input-image=MaxRGB --mbari-saliency-input-image=Raw --mbari-cache-size=2 --mbari-max-WTA-points=15 --mbari-max-evolve-msec=15000 --vc-type=OIC --use-random=false --test-mode=true |

|                                                                                                                                                                 |      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                                                                                                 |      | <pre>--oricomp-type=Steerable -- mbari-cache-size=2 --use- random=false --use-older-version=false --shape-estim- mode=ConspicuityMap --ior- type=ShapeEst</pre>                                                                                                                                                                                                                                                                                                                                                           |
| -mbari-benthic-video                                                                                                                                            | [no] | <p>Implements good choice of options to experiment with processing video from a moving camera traversing the sea bottom. EQUIVALENT TO: --mbari-saliency-dist=5 --mbari-tracking-mode=NearestNeighbor --mbari-saliency-input-image=DiffMean --mbari-segment-algorithm-input-image=MaxRGB --vc-type=O:5IC --use-random=false --test-mode=true --ori-interaction=None --oricomp-type=Steerable --mbari-cache-size=15 --use-random=false --use-older-version=false --shape-estim-mode=ConspicuityMap --ior-type=ShapeEst</p> |
| -mbari-midwater-video                                                                                                                                           | [no] | <p>Implements good choice of options to experiment with processing video from a moving camera traversing the midwater sea column. EQUIVALENT TO: --mbari-saliency-dist=5 --mbari-tracking-mode=KalmanFilter --mbari-saliency-input-image=DiffMean --mbari-segment-algorithm-input-image=MaxRGB --vc-type=O:5IC --use-random=false --test-mode=true --mbari-cache-size=10 --use-random=false --use-older-version=false --shape-estim-mode=ConspicuityMap --ior-type=ShapeEst</p>                                           |
| -mbari-mosaic-stills<br> <b>this option is not supported in pmbarivision</b> | [no] | <p>Implements good choice of options to experiment with still frames collected from a moving camera in mosaic form. EQUIVALENT TO: --mbari-saliency-dist=1 --mbari-tracking-mode=None --mbari-keep-boring-WTA-points=yes --boring-sm-mv=0.25 --mbari-save-non-interesting-events=yes --mbari-segment-algorithm-input-image=MaxRGB</p>                                                                                                                                                                                     |

|                                                                                                                                                                      |                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                                                                                                      |                | --vc-type=Variance --use-random=false --mbari-saliency-input-image=Raw<br>--mbari-cache-size=2 --mbari-max-WTA-points=25 --mbari-max-evolve-msec=15000                                                                                                                                                                                                                                                                                                                                                                                                   |
| -mbari-timelapse-stills<br><br> <b>this option is not supported in pmbarivision</b> | [no]           | Implements good choice of options to experiment with still frames collected from a stationary time-lapse camera. EQUIVALENT TO: --mbari-saliency-dist=1<br>--mbari-tracking-mode=NearestNeighbor --mbari-keep-boring-WTA-points=yes<br>--mbari-save-non-interesting-events=yes --mbari-segment-algorithm-input-image=MaxRGB<br>--mbari-saliency-input-image=Raw --mbari-cache-size=10 --qtime-decay=1.0<br>--vc-type=Variance --use-random=false --mbari-max-WTA-points=30<br>--mbari-max-evolve-msec=15000 --use-random=false --use-older-version=false |
| -mbari-timelapse-rover-stills                                                                                                                                        | [no]           | Implements good choice of options to experiment with time-lapse still frames collected from a benthic moving camera . EQUIVALENT TO: --mbari-saliency-dist=1<br>--mbari-tracking-mode=None --mbari-keep-boring-WTA-points=yes --qtime-decay=1.0<br>--mbari-save-non-interesting-events=yes --mbari-segment-algorithm-input-image=MaxRGB<br>--mbari-saliency-input-image=Raw --vc-type=O:5IC --use-random=false<br>--mbari-max-WTA-points=15 --mbari-max-evolve-msec=15000                                                                                |
| --mbari-max-WTA-points=<int>                                                                                                                                         | [20]           | Maximum number of winner-take-all points to find in each frame                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| --mbari-max-evolve-msec=<int>                                                                                                                                        | [500]          | Maximum amount of time in milliseconds to evolve the brain until stopping                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| --mbari-cache-size=<int>                                                                                                                                             | [30]           | The number of frames used to compute the running average.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| --mbari-tracking-mode=<KalmanFilter BoundingBox>                                                                                                                     | [KalmanFilter] | Tracking mode used to track events between frames                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |



|                                                                                                        |                  |                                                                                                                                                                               |
|--------------------------------------------------------------------------------------------------------|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| --mbari-segment-algorithm=<BinaryAdaptive AdaptiveThreshold BackgroundCanny HomomorphicCanny GraphCut> | [BinaryAdaptive] | Segmentation algorithm                                                                                                                                                        |
| --mbari-segment-algorithm-input-image=<MaxRGB Luminance>                                               | [MaxRGB]         | Segment algorithm input images type                                                                                                                                           |
| --mbari-segment-algorithm-se-type=<benthic midwater>                                                   | [benthic]        | Jerome Mariette's segmentation algorithm structure element test                                                                                                               |
| --mbari-segment-input-image=<Raw DiffMean>                                                             | [DiffMean]       | Saliency input image type                                                                                                                                                     |
| --mbari-mask-path=<file>                                                                               | []               | MaskPath: path to the mask image. <b>Important</b> this only masks the area to look for detections - it does not mask the area from the saliency computation.                 |
| --mbari-mask-xposition=<int>                                                                           | [1]              | MaskXPosition: x position of the mask point of reference                                                                                                                      |
| --mbari-mask-yposition=<int>                                                                           | [1]              | MaskYPosition: y position of the mask point of reference                                                                                                                      |
| --mbari-mask-width=<int>                                                                               | [1]              | MaskWidth: mask width                                                                                                                                                         |
| --mbari-mask-height=<int>                                                                              | [1]              | MaskHeight: mask height                                                                                                                                                       |
| --mbari-min-event-area=<int>                                                                           | [34]             | The minimum area an event must be to be candidate                                                                                                                             |
| --mbari-max-event-area=<int>                                                                           | [1000]           | The maximum area an event can be, to be candidate                                                                                                                             |
| --mbari-min-event-frames=<int>                                                                         | [1]              | The minimum number of frames an event must be to be candidate                                                                                                                 |
| --mbari-max-event-frames=<int>                                                                         | [-1]             | The maximum number of frames an event can be, to be candidate. Defaults to infinite                                                                                           |
| --mbari-saliency-dist=<int>                                                                            | [5]              | The number of frames to delay between saliency map computations Reduce this to improve your detection rate. Warning reducing this <b>will</b> increase your computation time. |

## Installing AVED Scripts

---

This page last changed on Mar 30, 2010 by [dcline](#).

### Installing AVED Scripts

There is a simple install script that will install the AVED scripts we have developed over the years. The install script will copy the scripts to the /usr/local/aved/scripts directory and setup your user paths. **You will need to be root user to run this.** Install with the following:

```
cd <full/path/to/mbarivision/scripts>
./install
```

This page last changed on May 28, 2013 by [dcline](#).

# Introduction

These are general administrative utilities for managing the MBARI Beowulf RAID. They are described here in case this is helpful for someone else.

## Utilities installed in /usr/local/bin

Install into crontab

```
#every 10 minutes, remount in case a mount point is intermittent
0-59/10 * * * * /bin/mount -a >/dev/null 2>&1
#on the first of every month, remove files older than 90 days
00 00 01 * * /usr/local/bin/cleanscratchfiles
#at midnight daily rebuild scratch in case new users added or permissions changed
59 23 * * * /usr/local/bin/buildscratch
#check disk space every hour and send root alert if getting too full
0 * * * * /usr/local/bin/diskalert
```

diskalert

```
#!/bin/sh
# set -x
# Shell script to monitor or watch the disk space
# It will send an email to $ADMIN, if the (free available) percentage of space is >= 90%.
# -----
# Set admin email so that you can get email.
ADMIN="root"

# set alert level 90% is default
ALERT=90

# Exclude list of unwanted monitoring, if several partions then use "|" to separate the partitions.
# An example: EXCLUDE_LIST="/dev/hdd1|/dev/hdc5"

EXCLUDE_LIST="tempest:/users|tempest:/tempbox:tornado:/Engineering|tornado:/vol/vol1/
ProjectLibrary|nanomia.private.net:/nanomiaRAID"
#
# ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
#
function main_prog() {
while read output;
do
#echo $output
usep=$(echo $output | awk '{ print $1}' | cut -d'%' -f1)
partition=$(echo $output | awk '{print $2}')
if [ $usep -ge $ALERT ] ; then
echo "Running out of space \"$partition ($usep)\" on server $(hostname), $(date)" |
mail -s "ALERT: Almost out of disk space $usep on $(hostname)" $ADMIN
fi
done
}

if [ "$EXCLUDE_LIST" != "" ] ; then
df -lH | grep -vE "^Filesystem|tmpfs|cdrom|${EXCLUDE_LIST}" | awk '{print $5 " " $6}' | main_prog
else
df -lH | grep -vE "^Filesystem|tmpfs|cdrom" | awk '{print $5 " " $6}' | main_prog
fi
```

buildscratch

```

! /bin/bash
#set -x
USERID=`id -u`
if [ $USERID != 0 ]; then
echo "You must be root to run buildscratch"
exit 0
fi

function dowork() {
echo "Building $1"

#remove all directories and recreate using file containing users
users=$(cat /etc/passwd |grep 20| cut -f1 -d ":")

pushd /mnt/scratch

#change permissions to include all users
chmod a+rwX /mnt/scratch

#change owner for new directories
for dir in $users
do
if [ ! -d $dir ]; then
echo "mkdir $dir"
mkdir $dir
fi
echo "chown $dir:users $dir"
chown $dir:users $dir
echo "chmod 0777 $dir "
chmod 777 $dir
done
chown aved:aved aved
popd

#refresh scratch space NFS mounts on all nodes
#cxexec '/bin/umount /mnt/scratch;/bin/mount /mnt/scratch'
}

for s in "/mnt/scratch" "/mnt/scratch2"
do
dowork $s
done

#rebuild video directories for capture
echo "Rebuilding video capture directories"
if [ ! -d /mnt/scratch/video ]; then
mkdir -p /mnt/scratch/video/capture
fi
chmod -Rf a+rwX /mnt/scratch/video

#rebuild directories for condor
if [ ! -d /mnt/scratch/condor ]; then
mkdir -p /mnt/scratch/condor -m=0755
fi
chown daemon:root /mnt/scratch/condor

```

## RAID format

Install xfsprogs

```
yum install xfsprogs
```

Reformat NSTOR RAID

```
fdisk /dev/sda  
pvcreate /dev/sda  
vgdisplay  
lvcreate -i2 -I4 -l 2198.7G -namedev AVEDRAID
```

Reformat RAID filesystem to XFS

```
mkfs.xfs -f -L "AVEDRAID" /dev/sda
```