

# DIVERSE Adaptable Display System (DADS) ©2003 - 2007

University Visualization and Animation Group

Virginia Tech Blacksburg, Virginia

and

Patrick Shinpaugh

Andrew Ray

Ron Kriz

## 1 - Overview

The DIVERSE Adaptable Display System (DADS) was initiated to realize a lower cost and more efficient CAVE system utilizing the DIVERSE API. Currently only Linux and IRIX platforms are supported by DADS though more platforms may be added in the future. DADS allows multiple computers to render to multiple displays in CAVE, RAVE, PowerWall, etc display configurations.

Unlike other solutions we do not use MPI or other parallel API. Instead we used DTK shared memory segments and TCP sockets to communicate between cluster console/server and cluster clients. Using DSOs (dynamic shared objects or plugins) and daemons (background processes) in conjunction with graphics hardware capable of framelock and swap sync, we were able to implement a highly efficient and relatively low cost system to drive the CAVE walls with full synchronization.

A client-server system was implemented where each client runs a daemon to receive and interpret messages. However, the servers are either applications or DSOs used by DTK/DPF/DGL applications. All messaging is handled by DTK shared memory segments created specifically for communicating between server and clients. A starter DSO was implemented such that when it is loaded it will gather the necessary information to replicate the process and pass it to the clients.

## 2 - Build and Installation

DADS now uses the GNU Autotools to greatly simplify the build and installation process.

The latest version of DADS can be downloaded from <http://sourceforge.net/projects/diverse/>. The tarball may have one of two different extensions. To decompress the tarball use one of the following commands.

For UNIXes (e.g. IRIX) with tar commands which do not support zip or bzip2 decompression, use the command

```
gunzip -c dads-1.3.0.tar.gz | tar xvf -
```

or

```
bunzip2 -c dads-1.3.0.tar.bz2 | tar xvf -
```

For machines with tar commands which do support zip or bzip2 decompression, use the command

```
tar xzvf dads-1.3.0.tar.gz
```

or

```
tar xjvf dads-1.3.0.tar.bz2
```

## 2.1 - Dependencies

Several additional functional packages are available as a part of the DADS tarball. Each functional package has its own set of dependencies.

| <i><b>Package</b></i>     | <i><b>Dependency</b></i>   |
|---------------------------|--|
| DADS components           | DTK, and DPF or DGL  |
| displayswitcher           | Qt, and Extron CrossPoint Matrix Switcher                          |
| dtk-swapServer            | DTK  |
| nvswapbarrier DSO         | DTK, and NVIDIA drivers supporting the GLX_NV_swap_group extension |
| dtk-masterServer programs | None   |

Table 1. Dependencies

## 2.2 - Build

To build DADS issue the following commands.

```
./configure  
make  
make install
```

There are several configuration options available in addition to the standard configure options.

| <i>Option</i>          | <i>Description</i>  | <i>Default</i>                                      |
|------------------------|---|---|
| --enable-debug         | Set environment to use debug mode   | Disabled  |
| --without-dads         | Build and install other features of the package without installing DADS   | With  |
| --with-displayswitcher | Build and install displayswitcher - requires Extron CrossPoint Matrix Switcher (default is no)  | Without   |
| --with-nvswapbarrier   | Build and install nvswapbarrier DSO for use with NVIDIA Quadro cards which support genlock and/or framelock (default is yes if NVIDIA GL headers are found, no otherwise) | With if NVIDIA headers are found, otherwise without |
| --with-dtkmasterserver | Build and install dtk-masterServer and dtk-masterControl programs for easier maintenance and resetting capabilities (default is yes if DTK is found, otherwise no)        | With if DTK is found, otherwise no                  |
| --with-dtkswapserver   | Build and install dtk-swapServer used to convert endianness of standard DTK shared memory segments (default is no)  | Without   |
| --prefix=PATH          | Specify the installation path   | /usr/local  |

Table 2. configure options

By default the CXXFLAGS will be set to use full optimization (-O3). If the --enable-debug option is passed to configure then the CXXFLAGS will be set to partial optimization (-O2) along with the specific flag to enable debugging symbols (typically -g).

## 2.2.1 - Example

For those who have never used configure scripts to build and install packages, this will give you a brief example. Let us say that you wish to install DADS, as well as the displayswitcher, and the dtk-masterServer programs and you want the installation directory to be /usr/local/diverse/dads-1.3.0. You would run the following command

```
./configure --with-displayswitcher --with-dtkmasterserver --prefix=/usr/local/diverse/dads-1.3.0
```

which will verify the existence of DTK and DPF (dtk-config and dpf-config must be in your PATH) as well as the existence of Qt (a product of Trolltech). The final output of running configure with the options shown would look similar to the output below.

```
-----
completed configure: DIVERSE Adaptable Display System
version: 1.3.0
-----
```

```
Architecture (DADS_ARCH):          LINUX
C++ Compiler (CXX):                g++
C++ Compiler Flags (CXXFLAGS):     -O3 -Wall -DDADS_ARCH_LINUX
C++ Linker Flags (LDFLAGS):
C/C++ preprocessor flags (CPPFLAGS):
Linker (LD):                       /usr/bin/ld
Installer (INSTALL):               /usr/bin/install -c
Make (MAKE):
Installation prefix (prefix):       /usr/local/diverse/dads-1.3.0
```

```
DIVERSE Toolkit (DTK):              2.4.14
DIVERSE Interface to OpenGL Performer (DPF):  2.4.2
Build dads components:              yes
Build dtk-masterServer programs:    yes
Build nvswapbarrier DSO:            yes
Build displayswitcher programs:     yes
Build dtk-swapServer programs:      no
```

You would then run make and make install to actually build and install the specified components.

## 2.3 - Installation

When 'make install' is run, the various components will be installed based upon the configure options given. The default \$PREFIX used for installation is /usr/local. The installation directories are dependent upon the functional packages that are built.

| <i>Package</i>    | <i>Installation Directory</i> |
|-------------------|-------------------------------|
| DADS/lib          | \$(PREFIX)/lib                |
| DADS/programs     | \$(PREFIX)/bin                |
| DADS/scripts      | \$(PREFIX)/bin                |
| DADS/contribDSOs  | dpf-config --dso-dir          |
| DADS/DSOs         | dtk-config --augment-dso-dir  |
| DADS/doc          | \$(PREFIX)/doc                |
| dtk-masterServer  | \$(PREFIX)/bin                |
| dtk-masterControl | \$(PREFIX)/bin                |
| dtk-swapServer    | \$(PREFIX)/bin                |
| displayswitchd    | \$(PREFIX)/bin                |
| displayswitcher   | \$(PREFIX)/bin                |

Table 3. package installation directories

The installation directories for DADS/contribDSOs and DADS/DSOs cannot be changed and will be dependent upon the values returned by dpf-config and dtk-config respectively.

### 3 - System Configuration

DADS has been tested on RedHat Linux 8.0, 9.0, and Fedora Core 2. There are a variety of configuration steps necessary to successfully configure DADS.

The latest nVidia drivers which implement nvidia-settings should be installed. After following the driver installation instructions, there are important modifications that need to be made to the XF86Config or xorg.conf file (which will hereafter be referred to as XF86Config). For framelock it is necessary to add a server flag option. To prevent blanking from occurring there are two server flags which need to be set. See the ServerFlags section of the XF86Config file in Appendix A.

The Monitor section of the XF86Config file should also be modified. Because we have 2 montiors (a projector and a common display through KVM) for each client, there are 2 Monitor sections. However, only 1 monitor is active at any time. We simply switch between them by switching their numbers and restarting the X server. This is only done when maintenance is required on one of the client machines. The Monitor section configured for the projectors will probably require that modelines be specified in order to have the vertical refresh frequency desired. An Xfree86 modeline generator conforming to the GTF standard is available at <http://gtf.sourceforge.net>.

The cluster clients must boot into run level 3. To force the machine to boot into runlevel 3 instead of runlevel 5, edit the /etc/inittab file. Look for the line similar to

```
id:5:initdefault:
```

and replace the 5 with a 3. This is necessary because run level 5 will not allow a remote process to instantiate graphic applications. The .xinitrc run control script is used to accomplish this task. If it is necessary to run xinit manually, be sure to run xinit as the root superuser as a background process. This step can be postponed until all installations have been performed and GUI interaction is no longer necessary.

```
[root@machine somedir]# xinit &
```

Please see Appendix A for a listing of the .xinitrc file for the framelock master and the framelock clients.

One machine is chosen as the framelock master and all others will be slaves to that master machine. The framelock master must have permission to access the displays of all of the framelock clients. From the framelock master machine call

```
xhost +client
```

where client is the hostname of the client for each framelock client. Each framelock client must have access to the framelock master display. From each framelock client call

```
xhost +server
```

where server is the hostname of the framelock master machine.

Use the nvidia-settings tool to set up the machines for framelock. It should be configured from the framelock master machine. Within the nvidia-settings tool, click on the framelock tab from the list on the left. Click the Add X Screen button and add the framelock master host first. The format should be

```
dads1:0.0
```

or

```
dads1.your.domain:0.0
```

where dads1 is the hostname of the framelock master and your.domain is your specific domain. Each client should then be added similarly. If there is an error adding any of the machines, verify that xhost has been set up properly for the framelock master and each of the clients.

After all of the machines have been added, click on the Enable Framelock button. This enables framelock which will synchronize vblanking on all of the framelocked machines. Looking on the back of the machines at the framelock connectors, the lights next to the cat 5 cables should be flashing. If they are not flashing, verify that you have completed each step properly. Also verify that the cat5 cables connected to the framelock ports are not also connected to any network ethernet ports (this could damage your graphics card).



## 4 - Environment Variables

There are several environment variables used with the cluster solution. Some are new and some are existing environment variables from DPF and DTK. All environment variables are set on the console/server machine and will be passed to the clients as needed.

**DTK\_DSO\_FILES** – From DTK. Use this in the console/server side to specify the DSOs to be loaded on the server.

**DPF\_DSO\_FILES** – From DPF. Use this in the console/server side to specify the DSOs to be loaded on the server.

**DGL\_DSO\_FILES** – From DGL. Use this in the console/server side to specify the DSOs to be loaded on the server.

**DADS\_DSO\_FILES** – Specify the DSOs to be loaded by all clients.

**DADS\_DSO\_FILES\_host** – Specify the DSOs to be loaded specifically for host where host is the base or alias of the hostname ( i.e. The host for [somehost.example.net](http://somehost.example.net) would be somehost ).

**DADS\_ENV** – Specify environment variables to be passed to all client machines. The environment variables must be set on the console/server machine prior to running diversify.

**DADS\_ENV\_host** – Specify environment variables to be passed to host where host is the base or alias of the hostname ( i.e. The host for [somehost.example.net](http://somehost.example.net) would be somehost ). The environment variables must be set on the console/server machine prior to running diversify.

**DADS\_CLIENTS** – a colon separated list of hosts which should all be running the dListener daemon.

**DADS\_SERVER** – the host name of the console/server which should not be running dListener. This environment variable should be set to the host name of the console/server machine.

**DADS\_SYNC\_SEGMENTS** – Specify a colon separated list of DTK shared memory segments to be synchronized through the navWrite and navRead DSOs. The shared memory segment values will be passed from the console/server machine using navWrite to all of the client machine through navRead.

**DADS\_DISPLAY\_host** – Specify the DISPLAY variable for each host if different than the :0.0 default.

**DADS\_TIMEOUT** – This timeout value (in seconds) determines the amount of time clients will be given to respond to the console machine. For larger or more complex models this value may need to be

increased. The default value is 20 seconds.

DADS\_CONFIG\_FILE – Allows the use of a configuration file.

PFSHAREDSIZE – The PFSHAREDSIZE environment variable is part of SGI's OpenGL Performer. It is mentioned here for troubleshooting purposes only. When using large or complex models with many textures it is sometimes necessary to increase the default PFSHAREDSIZE which specifies the number of bytes to set aside for the Performer shared arenas.

## 5 - Daemons

The daemons necessary for the operation of DADS are shown here.

### 5.1 - dListener

The daemon listens for messages on a shared memory segment. The messages are for starting a program, stopping a program, setting an environment variable, unsetting an environment variable, and checking the status of the listener daemon. The dListener must be running on each client machine or no action will be taken on that client. The application will timeout on the server and clients where it does spawn as the syncing requires communication from all clients.

### 5.2 - dRepeat

This daemon will repeat data from one shared memory segment to a different named shared memory segment. It sleeps between reads and writes to reduce CPU usage.

```
dRepeat will repeat shared memory segments such that other machines
can connect via remote shared memory. There are 2 methods of
repeating the shared memory segments with this daemon. The first
is using --segment and providing a list of segments where the
repeated segments will each have the basename with a suffix of
repeat added to the ending. (i.e. dRepeat --segment head wand
will result in repeated shared memory segments of headrepeat
and wandrepeat).
```

```
The second method is to use --from segment --to segment pairs
where the from segment will be repeated with the to segment
name. Within each pair the from segment must precede the to
segment.
```

**Options:**

|                          |                          |   |
|--------------------------|--------------------------|---|
| <code>--segment</code>   | <code>segmentlist</code> | This will repeat each segment in the segment list with a suffix of repeat (i.e. dRepeat --segment head wand will result in repeated shared memory segments of headrepeat and wandrepeat). |
| <code>-s</code>          | <code>segmentlist</code> |   |
| <code>--from</code>      | <code>segment</code>     | Must be paired with a --to segment. Specify the existing segment to be repeated.  |
| <code>-f</code>          | <code>segment</code>     |   |
| <code>--to</code>        | <code>segment</code>     | Must be paired with a --from segment. Specify the name of the repeated segment.   |
| <code>-t</code>          | <code>segment</code>     |   |
| <code>-h   --help</code> |                          | Shows this usage/help information.  |

## 6 - Applications

### 6.1 - dListener-stop

This application is used to kill the dListener daemon. It has no options. If a dListener is running on the machine, dListener-stop should cause the dListener to stop.

### 6.2 - dKiller

This application is used to kill processes on client machines which are no longer functioning properly. The dListener daemon must be running on the client machine for this application to function properly.

```
dKiller is used to kill processes which were started by the dListener daemon running on a client machine. Processes can be killed on a specific machine or all machines defined in the DADS_CLIENTS environment variable. A specific program process can be killed or all processes started by the dListener daemon can be killed.
```

Options:

|                                |   |
|--------------------------------|---|
| hostname                       | If [hostname] is defined then a message will be sent to that client to kill the processes. Otherwise the message is sent to all hosts defined in the DADS_CLIENTS environment variable. |
| --kill progname<br>-k progname | Use the [kill program] option to kill a specific program which was started by the dListener where program is the name of the process to be killed.                                      |
| --killall<br>-a                | Use the [killall] option to kill all processes which were started by the dListener.   |
| -h   --help                    | Shows this usage/help information.  |

### 6.3 - dSetEnv

This application is used to set environment variables on client machines which are running the dListener daemon.

```
dSetEnv provides a means to set environment variables using the dListener daemon running on a remote machine. The environment variable can be set for all clients listed in the DADS_CLIENTS environment variable on the local machine or can be set using the computer/host name option for individual machines.
```

Options:

|                     |   |
|---------------------|---|
| -c hostname         | These options will set the name of the client |
| --computer hostname | as the hostname provided for which to set the |
| --hostname hostname | environment variable.                         |

|             |            |  |
|-------------|------------|--|
| -e          | name value | This mandatory option specifies the environment variable to be set with the specified value. |
| --env       | name value |  |
| -h   --help |            | Shows this usage/help information.   |

## 6.4 - dUnsetEnv

This application is used to unset environment variables on client machines which are running the dListener daemon.

dUnsetEnv provides a means to unset environment variables using the dListener daemon running on a remote machine. The environment variable can be unset for all clients listed in the DADS\_CLIENTS environment variable on the local machine or can be unset using the computer/host name option for individual machines.

### Options:

|             |          |   |
|-------------|----------|---|
| -c          | hostname | These options will set the name of the client as the hostname provided for which to unset the environment variable. |
| --computer  | hostname |   |
| --hostname  | hostname |   |
| -e          | name     | This mandatory option specifies the environment variable to be unset.   |
| --env       | name     |   |
| -h   --help |          | Shows this usage/help information.  |

## 6.5 - dCreateShm

This application is used to create shared memory segments or messaging post offices. It can also be used to create the standard cave shared memory segments head, wand, joystick, and buttons.

dCreateShm is used to create shared memory or post office segments. Any combination of options is valid though it is the responsibility of the user to avoid duplicating names.

### Options:

|             |           |   |
|-------------|-----------|---|
| --shm       | name size | This will create a shared memory segment with the specified name and size.              |
| -s          | name size |   |
| --po        | name      | This will create a PostOffice with the name specified for use with messaging.           |
| -p          | name      |   |
| --cave      |           | This will create the standard shared memory segments head, wand, joystick, and buttons. |
| -c          |           |   |
| -h   --help |           | Shows this usage/help information.  |

## 6.6 - dExec

This application can be used to start applications remotely on a client running the dListener daemon.

dExec provides a means to initiate programs/processes using the dListener daemon running on a remote machine. The program/process can be started for all clients listed in the DADS\_CLIENTS environment variable on the local machine or can be set using the computer/host name option for individual machines.

Options:

|             |           |   |
|-------------|-----------|---|
| -c          | hostname  | These options will set the name of the client   |
| --computer  | hostname  | as the hostname provided for which to start     |
| --hostname  | hostname  | the program/process.                            |
| -p          | name args | This mandatory option specifies the name of the |
| --program   | name args | program/process to be started and its arguments |
| -h   --help |           | Shows this uasage/help information.             |

## 7 - DSOs

Most of these DSOs are necessary for the proper operation of the DADS system with the exception of dPerformance and dHideCursor. Although not necessary they do play important roles within the DADS system.

### 7.1 - starter

The starter DSO will connect to the message post offices of all clients defined in the DADS\_CLIENTS environment variable and send messages telling all clients running dListener to start the application. The message will consist of the same command line arguments and will also pass the DADS\_DSO\_FILES to the client as DPF\_DSO\_FILES. The starter DSO is required for the cluster solution to work properly. It is the method by which the client machines running dListener are told to start a program.

### 7.2 - death

The death DSO is used to exit the client applications normally when the console/server application has been requested to exit normally. This should be included in the DADS\_DSO\_FILES environment variable used on the client machine

### 7.3 - navWrite

The navWrite DSO uses a software syncing solution to synchronize the location and head tracker position with clients running the navRead DSO. Must be used in conjunction with the navRead DSO or another navigation DSO implemented to synchronize similarly to the navRead DSO.

### 7.4 - navRead

The navRead DSO uses a software syncing solution to retrieve the location and head tracker position with the console/server machine running the navWrite DSO. Must be used on the client in conjunction with a server running the navWrite DSO or another navigation DSO implemented to synchronize similarly to the navWrite DSO.

### 7.5 - nvswapbarrier

This DSO provides swap barrier support for NVidia graphics cards supporting framelock and the NV\_swap\_group extension (i.e. NVidia Quadro FX 3000G). Only works in mono mode. The NVIDIA drivers for the Quadro FX 3000G are beta quality and only available through the PNY Technologies website (<http://www.pny.com>). In stereo mode the displays will not provide stereo at a maintainable refresh rate and would likely result in a very uncomfortable immersive experience.

## 7.6 - dHideCursor

This DSO hides the cursor within the window associated with the application while it is running. This DSO only works with DPF.

## 7.7 - dPerformance

This DSO logs performance (frames per second/FPS) to a file called fps.log which is stored in the /tmp directory. The performance data is logged every 60 frames. The total number of frames and the average fps will be logged when the application exits.



## 8 - contribDSOs

The contribDSOs provide examples of DSOs that need to be created for a specific environment. The DSOs mentioned below are for the 4 wall CAVE at Virginia Tech. You should be able to modify the existing source code to run DADS on your CAVE, RAVE, or PowerWall.

### 8.1 - vtCaveClusterGroup

This DSO provides all of the functionality necessary to run the DADS system from the console machine with stereo support. You should modify the server and client host names to match your CAVE/RAVE system. This DSO only works with DPF.

#### 8.1.2 - vtCaveMonoClusterGroup

This DSO provides all of the functionality necessary to run the DADS system from the console machine without stereo support (mono). You should modify the server and client host names to match your CAVE/RAVE system. This DSO only works with DPF.

### 8.2 - vtCaveClusterClientGroup

This DSO provides all of the functionality necessary for the clients to run properly in the cave with the exception of the display. Displays must be specified for each client. Includes xkeyboardMouseInput, caveDTKInput, setHeadView, caveSim, navRead, death, toggleObjectsGroup, toggleScreenFrame, and debugHeadsUp DSOs. This DSO only works with DPF.

### 8.3 - vtCaveDisplayFront

The standard display provides fullscreen 1280 x 1024 resolution at 0 degree offset from center with stereo support. This DSO only works with DPF.

### 8.4 - vtCaveDisplayFloor

This display provides fullscreen 1280 x 1024 resolution at -90 degree rotation about the x axis with stereo support. This DSO only works with DPF.

### 8.5 - vtCaveDisplayLeft

This display provides fullscreen 1280 x 1024 resolution at -90 degree rotation about the z axis with stereo support. This DSO only works with DPF.

### 8.6 - vtCaveDisplayRight

This display provides fullscreen 1280 x 1024 resolution at 90 degree rotation about the x axis with stereo support. This DSO only works with DPF.

### 8.3 - vtCaveMonoDisplayFront

The standard display provides fullscreen 1280 x 1024 resolution at 0 degree offset from center without stereo support (mono). This DSO only works with DPF.

### 8.4 - vtCaveMonoDisplayFloor

This display provides fullscreen 1280 x 1024 resolution at -90 degree rotation about the x axis without stereo support (mono). This DSO only works with DPF.

### 8.5 - vtCaveMonoDisplayLeft

This display provides fullscreen 1280 x 1024 resolution at -90 degree rotation about the z axis without stereo support (mono). This DSO only works with DPF.

### 8.6 - vtCaveMonoDisplayRight

This display provides fullscreen 1280 x 1024 resolution at 90 degree rotation about the x axis without stereo support (mono). This DSO only works with DPF.

## 9 - contribPrograms

The programs and servers are not a part of DADS but are very useful in our CAVE.

### 9.1 - dtk-masterServer

dtk-masterServer is a daemon which listens for messages from various programs to reset dtk-server and dtk-server services (i.e. is900, wingmanJoystick, etc) as well as dListener daemons. This was created to minimize security risks from the scripts and programs used to resetting the DADS and CAVE systems which required the direct use of the root password.

#### Options:

--shutdown | -s      Use this option to shutdown the dtk-masterServer daemon.

#### Configuration:

dtk-masterServer uses a configurations file /etc/dtk-masterServer.conf which should only be viewable by root. Below is a list of options which can be set in the configuration file.

|                 |   |
|-----------------|---|
| port            | This is the port on which the server will listen. Each client program which is run on other machines should be running on the same port. The default is 65535.      |
| hosts_accept    | A colon separated list of hostnames from which the server will accept commands. Default is localhost.   |
| hosts_reset     | A colon separated list of hostnames to which the server will send reset commands. There is no default.  |
| start_dtkserver | A boolean value specifying whether the dtk-server should be started when a reset occurs. Acceptable values are yes, no, true, or false. The default value is false. |
| start_dListener | A boolean value specifying whether the dListener should be started when a reset occurs. Acceptable values are yes, no, true, or false. The default value is false.  |
| dtk_services    | A colon separated list of service names (i.e. is900, wingManJoystick, etc) which should be started when a reset occurs. There is no default.                        |
| logging         | A boolean value specifying whether warning and error messages should be logged. The default is false.   |
| logfile         | A filename (including path) where log information should be stored. The default is  |

|                 |   |
|-----------------|---|
|                 | /tmp/dtk-masterServer.log.  |
| apps_accept     | A colon separated list of programs/commands/servers which can be started through the dtk-masterServer. If not defined, none may be started locally. |
| services_accept | A colon separated list of dtk services which can be started through the dtk-masterServer. If not defined, none may be started locally.              |
| commands        | A colon separated list of commands to be run after starting servers/services.   |
| reset_commands  | A colon separated list of commands to be run after stopping any previously running servers/services but before starting any servers/services.       |

## 9.2 - dtk-masterControl

dtk-masterControl is a command which sends messages to the dtk-masterServer running on the local or another machine. It can be used to reset the specified machine to its default settings, add or remove services running on a machine, or stop all servers and services running on a machine.

When resetting or stopping a machine, all machines listed in the hosts\_reset parameter of the /etc/dtk-masterServer.conf file will be forced to reset or stop also.

Any service may be removed from running on the specified machine, but only authorized services can be started on the specified machine. These are listed in the /etc/dtk-masterServer.conf file with the services\_accept parameter. If services are added and deleted, deletions will occur first followed by additions. If services are added and a reset is also requested, the reset will occur first followed by the service additions

dtk-masterControl uses the /etc/dtk-masterserver.conf file to determine how it communicates and what it is capable of doing. The following options are used with dtk-masterControl.

```
--host | -H hostname The hostname of the machine which will receive the
                        commands from dtk-masterControl. The default is
                        the local machine.

--addsvc|-a service  Add/start the specified service if it is not
                        already running on the specified machine.

--delsvc|-d service  Remove the specified service if it is currently
                        running on the specified machine.

--reset | -R         Reset the servers running on the specified machine.
                        This will also reset all machines listed in the
                        hosts_reset parameter on the specified machine.

--stop | -S          Stops all dtk-server and dListener daemons running
                        on the specified machine. This will also stop all
                        machines listed in the hosts_reset parameter on the
```

|                               |  |
|-------------------------------|--|
|                               | specified machine.   |
| <code>--list   -l</code>      | List services that are currently running on the specified machine.                                       |
| <code>--paramlist   -L</code> | List services, apps, reset defaults from the configuration file for the specified machine.               |
| <code>--verbose   -v</code>   | Prints verbose output for each command. The default is to print "Success" or "Failure" for each command. |
| <code>--help   -h</code>      | Show this usage information.   |

### 9.3 - displayswitchd

This daemon listens for messages sent from the displayswitcher client program and then passes messages to the display switcher. As is, this program will only work with an Extron Electronics CrossPoint Matrix Switcher. It was designed specifically for the VT CAVE but if you have an Extron CrossPoint Matrix Switcher you find it to be of use through customization.

### 9.4 - displayswitcher

The displayswitcher is a simple GUI interface which communicates with the displayswitchd daemon to control the display switcher. As is, this program will only work with an Extron Electronics CrossPoint Matrix Switcher. It was designed specifically for the VT CAVE but if you have an Extron CrossPoint Matrix Switch you find it to be of use through customization.

### 9.5 - dtk-swapServer

The dtk-swapServer daemon is used to swap bytes in head, wand, joystick and buttons. We have a Linux box serving up the tracker data to other Linux boxes as well as to an SGI IRIX machine. The endianness of x86 machines and MIPS machines is not the same.

## 10 - Example

The dtk-server daemon should be running on all machines being used in the DADS system which includes the console and the client machines. The dListener daemon should be running on the client machines only.

The following script will run diversifly sub.pfb on the console machine and 4 cluster clients named dads1, dads2, dads3, and dads4.

```
export DISPLAY=localhost:0.0
export DPF_DSO_FILES=desktopCaveEmulateGroup:starter:navWrite
export DADS_CLUSTER_SERVER=dadsconsole
export DADS_CLUSTER_CLIENTS=dads1:dads2:dads3:dads4
export DADS_DSO_FILES=vtCaveClusterClientGroup:nvswapbarrier
export DADS_DSO_FILES_dads1=vtCaveDisplayFront
export DADS_DSO_FILES_dads2=vtCaveDisplayFloor
export DADS_DSO_FILES_dads3=vtCaveDisplayRight
export DADS_DSO_FILES_dads4=vtCaveDisplayLeft
diversifly sub.pfb
```

The script sets the DISPLAY to the localhost display screen 0. The DPF\_DSO\_FILES is used to set the DSOs to be loaded for the console/server machine (machine from which the script is run) to desktopCaveEmulateGroup (part of DPF), the starter DSO, and the navRead DSO. The name of the console/server machine is set using the DADS\_SERVER environment variable. This will be sent to each of the clients. The DADS\_CLIENTS environment variable is used to specify the client machines that will run the application. The DADS\_DSO\_FILES environment variable is used to specify the DSOs which will be loaded for all of the client machines. The DADS\_DSO\_FILES\_dads1 environment variable is used to specify the DSOs which will be loaded specifically for the dads1 client machine. The same is true for each of the DADS\_DSO\_FILES\_dadsx environment variables which provides a means of specifying individual DSOs to be loaded. Finally the application diversifly is called.

When diversifly is called, the starter DSO will send a message to each of the client machines listed in the DADS\_CLIENTS environment variable which will include the command line arguments for the call to the application (i.e. diversifly sub.pfb). The parameters of the DADS\_DSO\_FILES environment variable will be combined with the parameters for the DADS\_DSO\_FILES\_dadsx environment variable for each client machine and renamed to DPF\_DSO\_FILES. This DPF\_DSO\_FILES environment variable will be sent as part of the message to each client (i.e. dads1 will receive a DPF\_DSO\_FILES environment variable as DPF\_DSO\_FILES=vtCaveClusterGroup:vtCaveDisplayFront). The message also includes the user id and the current working directory.

The dListener running on each of the client machines checks for messages on the `COMMAND_SEGMENT` host shared memory segment where host is the base name of the machine hostname. The dListener daemon will interpret the message and will attempt to start the program. First it will fork a process, set the new process as a process leader, set the user id of the process to match the userid from the message, change to the current working directory, set the `DPF_DSO_FILES` environment variable and finally it will exec the commandline arguments passed to it. If everything goes as expected, all client machines will be running the application, each with its associated DSOs.

# Appendix A - Files

## /root/.xinitrc (framelock master)

```
xhost +dads2
xhost +dads3
xhost +dads4
/usr/bin/nvidia-settings --load-config-only &
xterm -g 80x24+50+50 &
```

## /root/.xinitrc (framelock clients)

```
xhost +dads1
xterm -g 80x24+50+50 &
```

## /etc/X11/XF86Config or

## /etc/X11/xorg.conf

# File generated by anaconda.

### Section "ServerLayout"

```
Identifier      "Anaconda Configured"
Screen          0  "Screen0"  0 0
InputDevice     "Mouse0"  "CorePointer"
InputDevice     "Mouse1"  "SendCoreEvents"
InputDevice     "Keyboard0" "CoreKeyboard"
```

EndSection

### Section "Files"

```
# The location of the RGB database. Note, this is the name of the
# file minus the extension (like ".txt" or ".db"). There is normally
# no need to change the default.
# Multiple FontPath entries are allowed (they are concatenated together)
# By default, Red Hat 6.0 and later now use a font server independent of
# the X server to render fonts.
```

```
RgbPath         "/usr/X11R6/lib/X11/rgb"
FontPath         "unix/:7100"
```

EndSection

### Section "Module"

```
Load  "dbe"
Load  "extmod"
Load  "fbdevhw"
Load  "glx"
Load  "record"
Load  "freetype"
Load  "type1"
```

EndSection

### Section "ServerFlags"

```
Option      "BlankTime"  "0"
Option      "NoPM"       "true"
Option      "AllowNonLocalXvidtune" "1"
```



EndSection

Section "InputDevice"

```
# Option "AutoRepeat" "500 5"
# when using XQUEUE, comment out the above line, and uncomment the
# following line
# Option "Protocol" "Xqueue"
# Specify which keyboard LEDs can be user-controlled (eg, with xset(1))
# Option "Xleds" "1 2 3"
# To disable the XKEYBOARD extension, uncomment XkbDisable.
# Option "XkbDisable"
# To customise the XKB settings to suit your keyboard, modify the
# lines below (which are the defaults). For example, for a non-U.S.
# keyboard, you will probably want to use:
# Option "XkbModel" "pc102"
# If you have a US Microsoft Natural keyboard, you can use:
# Option "XkbModel" "microsoft"
#
# Then to change the language, change the Layout setting.
# For example, a german layout can be obtained with:
# Option "XkbLayout" "de"
# or:
# Option "XkbLayout" "de"
# Option "XkbVariant" "nodeadkeys"
#
# If you'd like to switch the positions of your capslock and
# control keys, use:
# Option "XkbOptions" "ctrl:swapcaps"
#Option "XkbOptions" ""
Identifier "Keyboard0"
Driver "keyboard"
Option "XkbRules" "xfree86"
Option "XkbModel" "pc105"
Option "XkbLayout" "us" #Option "XkbVariant" ""
```

EndSection

Section "InputDevice"

```
Identifier "Mouse0"
Driver "mouse"
Option "Protocol" "IMPS/2"
Option "Device" "/dev/psaux"
Option "ZAxisMapping" "4 5"
Option "Emulate3Buttons" "no"
```

EndSection

Section "InputDevice"

```
Identifier "Mousel"
Driver "mouse"
Option "Device" "/dev/input/mice"
Option "Protocol" "IMPS/2"
Option "Emulate3Buttons" "no"
Option "ZAxisMapping" "4 5"
```

EndSection

Section "Monitor"

```
Identifier "Monitor0"
VendorName "Monitor Vendor"
ModelName "Projector"
HorizSync 15.0 - 130.0
VertRefresh 38.0 - 180.0
```

```

        Mode          "1280x1024"
        DotClock       192.54
        HTimings       1280 1368 1560 1864
        VTimings       1024 1032 1036 1075
        Flags          "Composite"
    EndMode
EndSection

Section "Monitor"
    Identifier        "Monitor1"
    VendorName        "Monitor Vendor"
    ModelName         "KVM Video Display"
    HorizSync         31.0 - 80.0
    VertRefresh       36.0 - 76.0
    Option            "dpms"
EndSection

Section "Device"
    # no known options
    #BusID
    Identifier        "NVIDIA Quadro 4 (generic)"
    Driver            "nvidia"
    VendorName        "NVIDIA Quadro 4 (generic)"
    BoardName         "NVIDIA Quadro 4 (generic)"
    Option            "Stereo" "3"
EndSection

Section "Screen"
    Identifier        "Screen0"
    Device            "NVIDIA Quadro 4 (generic)"
    Monitor           "Monitor0"
    DefaultDepth      24
    SubSection        "Display"
        Depth        16
        Modes         "1280x1024"
    EndSubSection
    SubSection        "Display"
        Depth        24
        Modes         "1280x1024" "1280x960" "1152x864" "1024x768" "800x600"
        "640x480"
    EndSubSection
EndSection

Section "DRI"
    Mode             0666
EndSection

```