#

NAME

vximtool –- A virtual and/or proxy display server for IRAF IIS protocol clients

SYNOPSIS

vximtool [*options* ...]
vximtool -proxy [*options ...] server [server* ...]

DESCRIPTION

VXIMTOOL is a image display server process much like *XIMTOOL*, except that all it normally does is respond to datastream requests to read and write to internal frame buffers maintained as arrays in memory. Multiple frame buffers and frame buffer configurations are supported. It can be used to debug IIS protocol client programs by printing out the protocol packets recieved, or can simply be used as a dummy server in cases where no image display is really needed. By enabling the *-proxy* flag the server can also be used to repeat the datastream requests to a list of other servers, effectively splitting the image display to a number of other servers. See below for details on using the program in this manner.

The *-verbose* flag will log all datastream requests to the stderr, otherwise the task runs silently except for error messages. The process is terminated with an EOF on the stdin, if the task is intended to be run as a background job the *-background* flag should be enabled to shut off the check of the stdin and avoid a "waiting for tty input" message that would suspend the background job. Frame buffers are maintained as rasters in memory, up to four frames may currently be defined. If it not necessary to keep the displayed images in memory (i.e. the client will not need to read back any subrasters such as when doing overlay graphics), the *-noraster* flag can be used to disable the memory rasters. In this case any image readback will return an array of zero values.

The default frame buffer size is 512x512, with 2 frames defined initially. The *-config* and *-nframes* flags can be used to change the startup frame buffers used. The frame buffer configuration table is normally taken to be /usr/local/lib/imtoolrc but can be overridden by using an IMTOOLRC environment variable, a *$HOME/.imtoolrc* file, or the *-imtoolrc* command-line flag respectively. The format of the frame buffer configuration file is

configno nframes width height [extra fields]

e.g.

```
1  2  512  512
2  2  800  800
3  1 1024 1024        # comment
```

and so on. At most 128 frame buffer sizes may be defined, each configuration may define up to 4 frames, configuration numbers need not be sequential.

OPTIONS

-background
The task will be run in the background so don’t listen to STDIN for an EOF to quit gracefully.

-config *num*
Initial frame buffer configuration number. The default value is 1, indicating a 512x512 frame buffer with 2 frames. See below for information on the frame buffers.

-fifo *pipe*
Specifies the name of the fifo pipe to be used, the *i* and *o* suffixes will be added automatically. The default pipe names will be /dev/imt1i (input pipe) and /dev/imt1o (output pipe).

-fifo_only
If set, only fifo pipes will be used for communication with a client program, sockets will be disabled.

-help Print a summary of command line options to the screen.

-interactive
Allow cursor value strings to be typed into the STDIN in response to cursor read requests from the client.

-imtoolrc *file*
Specifies the frame buffer configuration file to be used. See below for information on frame buffers.

-inet_only
If set, only inet sockets will be used for communication with a client program, fifo pipes and unix sockets will be disabled.

-noraster
If set no client data will be stored in memory, and image readback will return an array of zeroes unless used as a proxy server.

-nframes *num*
Specifies the number of frame buffers to configure at startup. By default there will be 2 frames available, a maximum of 4 frames are allowed.

-port_only
Same as *-inet_only* option. If set, only inet sockets will be used for communication with a client program.

-port *num*
Set the primary port to listen on for client connections. The default is 5137 except when used as a proxy server in which case the default is 5136.

-proxy
Run as a proxy server by repeating client requests to each of the *server* remote server arguments. See the section on proxy usage below for details.

-verbose
If set, print information about all IIS packets headers recieved to the STDERR stream.

-unix *name*
Specifies the unix domain socket name to use. A "%d" in the filename will be replaced with the user id.

-unix_only
If set, only unix domain sockets will be used for communication with a client program, inet sockets and fifos will be disabled.

CLIENT CONNECTIONS

VXImtool allows clients to connect in any of the following ways:

fifo pipes
The traditional approach. The default global /dev/imt1[io] pipes may be used, or a private set of fifos can be specified using the *-fifo* command line argument. Values should be specified as the root pathname to a pair of fifo pipes whose last character is 'i' or 'o', these characters will be added automatically when opening the pipes. For example, to use the default pipes the path would be specified as simply "/dev/imt1". A value of "none" disables this connection.

tcp/ip sockets
Clients connect via a tcp/ip socket. The default port is *5137*, or a custom port may be specified using the *-port* command line switch. This permits connecting to the server over a remote network connection anywhere on the Internet. When used as a proxy server the default port number is 5136 so as to avoid a conflict with an XImtool running on the same machine, other connections are disabled unless otherwise overridden by a command line option following the *-proxy* flag to reset

the connection. A port number of 0 (zero) disables this connection.

unix domain sockets

Like a tcp/ip socket, but limited to a single host system. Usually faster than a tcp/ip socket, and comparable to a fifo. By default each user gets their own unix domain socket, so this option allows multiple users to run ximtools on the same host without having to customize things. The default value is "/tmp/.IMT%d", other sockets may be defined using the *-unix* command line switch. Legal values should be specified as a filename to be used for the socket, up to two "%d" fields are allowed and will be replaced by the userid. An empty string value disables this connection.

By default *vximtool* listens simultaneously for client connections on all three types of ports. Clients may connect simultaneously by different means allowing up to three different displays to be loading at the same time into different frames.

COMMUNICATIONS PROTOCOL

Clients communicate with *vximtool* using a protocol developed originally for IIS (International Imaging Systems) Frame Buffer hardware, the so-called "IIS protocol"; other more modern protocols will likely be supported in the future. The IIS protocol is basically a command packet stream with a header describing the operation to be performed (select frame, load display, read cursor, etc), and an optional data packet containing e.g. pixels. It is beyond the scope of this document to describe fully the details of the protocol; interested users should contact *iraf@noao.edu* for further information or see the *A Reference Guide for the IRAF Client Display Library* for details.

PROXY USAGE

The *vximtool* may also be used as a proxy server by enabling the *-proxy* command line flag. In this mode the program acts as a relay for the IIS datastream packets, sending image data, frame requests, etc. to a list of other servers specified on the command line. The effect of this is to allow a client to display to this program which then re-displays to each of the other named servers. A maximum of 8 servers may be named, they may be either on the local host or a remote machine and connections can be established using either fifos or sockets. See below for details on how to specify the server connection.

The current implementation has a few restrictions users should keep in mind:

Cursor and image readback are done by sending the request *only* to the first server named on the command line. This is done to avoid forcing a cursor mode on all servers which cannot be terminated when a response is received from only one server, and means that the first server named should be the one used to control interactive sessions. The remaining servers however can still respond to cursor requests from other applications connected to that server on another channel.

All named servers must be running prior to starting the proxy server. The connection to the remote servers is established when this task is first run and if no server is running that connection will be ignored. The task will exit if no remote servers can be found for display.

Any connected server that shuts down while the proxy server is running is likely to cause the program to crash on the next display.

SPECIFYING THE REMOTE SERVER CONNECTION

The syntax for the server argument used in proxy mode is as follows:

<domain> : <address>

where <domain> is one of **inet** (internet tcp/ip socket), **unix** (unix domain socket) or **fifo** (named pipe). The form of the address depends upon the domain, as illustrated in the examples below.

inet:5137 Connect to port 5137 on the local host.

inet:5137:foo.bar.edu	Client connection to port 5137 on internet host *foo.bar.edu*. The dotted (numeric) form of address may also be used.
unix:/tmp/.IMT212	Unix domain socket with the given pathname. IPC method, local host only.
fifo:/dev/imt1i:/dev/imt1o	FIFO or named pipe with the given pathname. IPC method, local host only. Two pathnames are required, one for input and one for output, since FIFOs are not bidirectional.

The address field may contain up to two "%d" fields. If present, the user's UID will be substituted (e.g. "unix:/tmp/.IMT%d").

EXAMPLES

1) Run the vximtool, logging output to the file named "spool":

```
% vximtool -verbose >& spool
```

2) Run the vximtool in the background, connect only on unix sockets, no output:

```
% vximtool -b -unix_only &
```

3) Don't store images in memory, start with initial 1024x1024 frame buffer:

```
% vximtool -noraster -config 3
```

4) Run the vximtool in the background, taking cursor input from a file:

```
% vximtool -i < cursor_file &
```

5) Run as a proxy server, displaying to both a local and a remote XImtool, log output to stderr:

```
...start all XImtool servers to be used
% vximtool -v -proxy unix:/tmp/.IMT%d inet:5137:pisces
Connected to server on unix:/tmp/.IMT359
Connected to server on inet:5137:pisces
Open to accept input on inet: port 5136
```

To display to the proxy server the client should connect to the vximtool on *inet:5136*, the proxy in turn will relay data to the servers listed on the command line. Note that in this last example any cursor requests from the client connected to the vximtool will be handled by the ximtool connected on a unix socket on the local machine (i.e. the first server argument). The remote ximtool on the machine pisces could simultaneously be used by another client on that machine connected by some other means. In this case it will only display images coming from the *vximtool* proxy, but could be used in an interactive IRAF session running on pisces to examine the images.

SEE ALSO

ximtool(1)
A Reference Guide for the IRAF Client Display Library (CDL)

COPYRIGHT

Copyright(c) 1986 Association of Universities for Research in Astronomy Inc.