Sun Ray Server Software 1.3
Man Pages

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303-4900 U.S.A.
650-960-1300

Part No. 806-7721-10
July 2001, Revision A

Send comments about this document to: docfeedback@sun.com

Contents

Preface

Both novice users and those familiar with the Sun Ray software can use online man pages to obtain information about the system and its features. A man page is intended to answer concisely the question “What does it do?” In general, man pages comprise a reference manual. They are not intended to be a tutorial.

Format

The following is a generic format for man pages. The man pages of command or file generally follow this order, but include only needed headings. For example, if there are no bugs to report, there is no BUGS section. See the `man`(1) command for more information about man pages in general.

NAME	This section gives the names of the commands or functions documented, followed by a brief description of what they do.
SYNOPSIS	This section shows the syntax of commands or functions. When a command or file does not exist in the standard path, its full path name is shown. Options and arguments are alphabetized, with single letter arguments first, and options with arguments next, unless a different argument order is required. The following special characters are used in this section:

`[ ]`	Brackets. The option or argument enclosed in these brackets is optional. If the brackets are omitted, the argument must be specified.
...	Ellipses. Several values may be provided for the previous argument, or the previous argument can be specified multiple times, for example "`filename...`".
|	Separator. Only one of the arguments separated by this character can be specified at one time.
`{ }`	Braces. The options and/or arguments enclosed within braces are interdependent, such that everything enclosed must be treated as a unit.

PROTOCOL	This section occurs only in subsection 3R to indicate the protocol description file.
DESCRIPTION	This section defines the functionality and behavior of the service. Thus it describes concisely what the command does. It does not discuss OPTIONS or cite EXAMPLES. Interactive commands, subcommands, requests, macros, functions and such, are described under USAGE.
IOCTL	This section appears on pages in Section 7 only. Only the device class which supplies appropriate parameters to the `ioctl`(2) system call is called `ioctl` and generates its own heading. `ioctl` calls for a specific device are listed alphabetically (on the man page for that specific device). `ioctl` calls are used for a particular class of devices all of which have an `io` ending, such as `mtio`(7I)
OPTIONS	This lists the command options with a concise summary of what each option does. The options are listed literally and in the order they appear in the SYNOPSIS section. Possible arguments to options are discussed under the option, and where appropriate, default values are supplied.
OPERANDS	This section lists the command operands and describes how they affect the actions of the command.

OUTPUT	This section describes the output – standard output, standard error, or output files – generated by the command.
RETURN VALUES	If the man page documents functions that return values, this section lists these values and describes the conditions under which they are returned. If a function can return only constant values, such as 0 or –1, these values are listed in tagged paragraphs. Otherwise, a single paragraph describes the return values of each function. Functions declared void do not return values, so they are not discussed in RETURN VALUES.
ERRORS	On failure, most functions place an error code in the global variable `errno` indicating why they failed. This section lists alphabetically all error codes a function can generate and describes the conditions that cause each error. When more than one condition can cause the same error, each condition is described in a separate paragraph under the error code.
USAGE	This section lists special rules, features and commands that require in-depth explanations. The subsections listed below are used to explain built-in functionality: Commands Modifiers Variables Expressions Input Grammar
EXAMPLES	This section provides examples of usage or of how to use a command or function. Wherever possible a complete example including command line entry and machine response is shown. Whenever an example is given, the prompt is shown as `example%` or if the user must be superuser, `example#`. Examples are followed by explanations, variable substitution rules, or returned values. Most examples illustrate concepts from the SYNOPSIS, DESCRIPTION, OPTIONS and USAGE sections.

ENVIRONMENT VARIABLES	This section lists any environment variables that the command or function affects, followed by a brief description of the effect.
EXIT STATUS	This section lists the values the command returns to the calling program or shell and the conditions that cause these values to be returned. Usually, zero is returned for successful completion and values other than zero for various error conditions.
FILES	This section lists all filenames referred to by the man page, files of interest, and files created or required by commands. Each is followed by a descriptive summary or explanation.
ATTRIBUTES	This section lists characteristics of commands, utilities, and device drivers by defining the attribute type and its corresponding value. See `attributes`(5) for more information.
SEE ALSO	This section lists references to other man pages, in-house documentation and outside publications.
DIAGNOSTICS	This section lists diagnostic messages with a brief explanation of the condition causing the error.
WARNINGS	This section lists warnings about special conditions which could seriously affect your working conditions. This is not a list of diagnostics.
NOTES	This section lists additional information that does not belong anywhere else on the page. It takes the form of an aside to the user, covering points of special interest. Critical information is never covered here.
BUGS	This section describes known bugs and wherever possible, suggests workarounds.

NAME

`auth.props` - Sun Ray authentication daemon configuration file.

SYNOPSIS

`/etc/opt/SUNWut/auth.props`

DESCRIPTION

The `auth.props` file contains the Sun Ray Authentication Manager's configuration properties. Changes to many of these properties are not supported and should not be set to other than the default values.

OPTIONS

The following options are supported:

`acceptRedirectToken=`*boolean*

After the Authentication Manager authenticates a token, it might discover that a redirect is required. The originating Authentication Manager sends the authenticated token to a target Authentication Manager. Setting this property to `false` causes the target Authentication Manager to distrust the authenticated token and re-authentication is required.

The default is set to `false` for higher security. Set to `true` if you want to accept authenticated redirects.

`adminConfigFile=`*filename*

This file contains the administrative database configuration information.

`allowAnnotations=`*boolean*

UNSUPPORTED. When set true, any application can connect from any IP address and annotate a session. Annotations are restricted to keywords prefixed by "x_". Values are not restricted.

`allowFWLoad=`*boolean*

Specifies whether or not the `utload` command is allowed to download firmware to appliances connected to this Authentication Manager.

`allowLANConnections=`*boolean*

UNSUPPORTED. When set true the appliance connections are allowed from `localhost` as well as from non-Sun Ray interconnect interfaces.

`cbport=`*portNumber*

UNSUPPORTED. The Authentication Manager listens on this port for connections from the `utsessiond` daemon and other programs, such as `utload`.

cbtimeout=*seconds*	UNSUPPORTED. Specifies the read timeout in seconds for programs that connect to the cbport.
controllers=*maximum*	UNSUPPORTED. Specifies the maximum number of spare threads that are available for handling new connections from applications such as utload(1M).
enableGroupManager=*boolean*	UNSUPPORTED. Flag to turn on the group manager function.
enableLoadBalancing=*boolean*	Flag to turn on group manager load balancing.
enableMulticast=*boolean*	UNSUPPORTED. Flag to enable / disable use of multicast in group manager. If disabled, group manager will use broadcast.
forceSessionLocation=*boolean*	UNSUPPORTED. Flag to force use of sessionHost and sessionPort settings from this file regardless of the various authentication modules.
gmDebug=*level*	UNSUPPORTED. Group manager debugging level.
gmKeepAliveInterval=*seconds*	UNSUPPORTED. The group manager uses this as the time in seconds between broadcast keepalive messages.
gmport=*port*	UNSUPPORTED. The group manager uses this port to send and receive keepalive / discovery messages from other Authentication Managers.
gmSignatureFile=*file*	The group manager can "sign" messages to other group managers based on the contents of a signature file. Other group managers with the same signature file contents are "trusted". To be usable, the file must be owned by 'root' and must not be readable, writable, or executable by anyone else; it must contain at least 8 bytes, at least two of which are letters and at least one which is a non-letter printable character.
log=*filename*	UNSUPPORTED. This option specifies a file that contains the log messages.

`cbtimeout=`*seconds*	UNSUPPORTED. Specifies the read timeout in seconds for programs that connect to the cbport.
`controllers=`*maximum*	UNSUPPORTED. Specifies the maximum number of spare threads that are available for handling new connections from applications such as `utload`(1M).
`enableGroupManager=`*boolean*	UNSUPPORTED. Flag to turn on the group manager function.
`enableLoadBalancing=`*boolean*	Flag to turn on group manager load balancing.
`enableMulticast=`*boolean*	UNSUPPORTED. Flag to enable / disable use of multicast in group manager. If disabled, group manager will use broadcast.
`forceSessionLocation=`*boolean*	UNSUPPORTED. Flag to force use of `sessionHost` and `sessionPort` settings from this file regardless of the various authentication modules.
`gmDebug=`*level*	UNSUPPORTED. Group manager debugging level.
`gmKeepAliveInterval=`*seconds*	UNSUPPORTED. The group manager uses this as the time in seconds between broadcast keepalive messages.
`gmport=`*port*	UNSUPPORTED. The group manager uses this port to send and receive keepalive / discovery messages from other Authentication Managers.
`gmSignatureFile=`*file*	The group manager can "sign" messages to other group managers based on the contents of a signature file. Other group managers with the same signature file contents are "trusted". To be usable, the file must be owned by 'root' and must not be readable, writable, or executable by anyone else; it must contain at least 8 bytes, at least two of which are letters and at least one which is a non-letter printable character.
`log=`*filename*	UNSUPPORTED. This option specifies a file that contains the log messages.

logAddTimeStamp=*boolean*	UNSUPPORTED. Add your own timestamp to syslog messages. This may be appropriate for debugging or in cases where a remote syslog server is being used and higher resolution timestamps are required.
logFacility=*value*	The logFacility can be one of the following: kern, user, mail, daemon, auth, syslog, lpr, news, uucp, cron, local0, local1, local2, local3, local4, local5, local6, local7
Log files	Log priorities for different utauthd message classes can be one of the following: emerg, alert, crit, err, warning, notice, info, debug, OFF. The message classes are: logPriClientError=*value* logPriDebug=*value* logPriNotice=*value* logPriWarning=*value* logPriConfigError=*value* logPriUnexpectedError=*value*
maxStarting=*maximum*	UNSUPPORTED. Specifies the maximum number of threads that can be simultaneously initiating a session. Additional threads wanting to start or verify a session wait for previous threads to finish starting or verifying a session.
moduleDif=*directorName*	UNSUPPORTED. Specifies the location of the authentication modules.
multicastTTL=*integer*	UNSUPPORTED. Time-to-live parameter for forwarding multicast packets. If set above one, keepalive messages can pass through routers.
noClaimSleepTime=*seconds*	UNSUPPORTED. The amount of time in seconds to sleep after a token has been offered to all of the authentication modules and before notifying the appliance that the authentication failed.

`policy=`*filename*	Specifies the location of the authentication policy specification.
`port=`*portNumber*	The `utauthd` daemon listens on this port for connections from Sun Ray appliances.
`remoteSelect=`*boolean*	If true, the remote server selection option of the `utselect`(1) command is enabled by default.
`reportAllDesktopEvents=`*boolean*	UNSUPPORTED. When true, all desktop events are reported instead of being filtered to just those events that change the "exists" state of the appliance.
`restrictSunrayIfs=`*boolean*	UNSUPPORTED. Flag to restrict communication between group managers on different hosts to travel over Sun Ray network interfaces. If false, group managers communicate over all interfaces.
`selectAtLogin=`*boolean*	If true, activates a `utselect -L` GUI allowing the user to select a Sun Ray server before logging into CDE. If only one server is available, the GUI exits automatically. Refer to the `utselect` man page for more information on the `-L` option.
`sessionHost=`*hostname*	UNSUPPORTED. Specifies the host name of the server that is running the default `utsessiond` for this Authentication Manager.
`sessionPort=`*portNumber*	UNSUPPORTED. Specifies the port number of the server that is running the default `utsessiond` for this Authentication Manager.
`sessionTypesFile=`*filename*	Specifies a file that contains mappings from session types to the associated session startup and shutdown commands.
`smtimeout=`*seconds*	UNSUPPORTED. Specifies the read timeout in seconds for reading messages from the `utsessiond` daemon.
`termAddrIsSecret=`*boolean*	UNSUPPORTED. When true, the IP address and port of appliances are not reported in the dynamic status information provided on port cbport in response to the string.
`terminateEnable=`*boolean*	UNSUPPORTED. Enables experimental code in `utauthd`.

timeout=*seconds*	UNSUPPORTED. Appliances are required to send a message to the Authentication Manager at least once every time period specified by *seconds*.
tokenDir=*directory*	UNSUPPORTED. Specifies a directory that contains the mappings from logical token names to session identifiers. The persistent storage of these mappings allows the utauthd daemon to recover its state after restarting. This state is reset on reboot of the system.
token.equiv=*filename*	UNSUPPORTED. Specifies a file that contains mappings from one raw token name to another.
useLocalPolicy=*boolean*	In a failover environment this is set to false to provide a global group policy (which is extracted from the datastore). Default for a single system is true. When configured for a failover group, the entry is normally false. When set true, only local policies will be used and the global policy entry in the LDAP database is ignored. Checked by utpolicy -G.
workers=*maximum*	UNSUPPORTED. Specifies the maximum number of spare threads that are available for handling new connections from Sun Ray appliances.

FILES

The following files are used:

/etc/init.d/utsvc	This is the system startup script that invokes the daemon /opt/SUNWut/utsessiond. The session manager performs the actual session switching function.

ATTRIBUTES

See attributes(5) for descriptions of the following attributes:

Attribute Type	Attribute Value
Availability	SUNWutr

SEE ALSO

utauthd(1M), utpolicy(1M), utsessiond(1M), utselect(1)

NAME

`sunray` - Sun Ray virtual device driver.

SYNOPSIS

`/dev/sunray`

DESCRIPTION

The `/dev/sunray` file refers to a pseudo-device driver that provides frame-buffer compatible information for configuring the `Xsun`(1) X server. The `sunray` driver's only function is to properly respond to the VIS_GETIDENTIFIER `ioctl`(2).

ATTRIBUTES

See `attributes(5)` for descriptions of the following attributes:

Attribute Type	Attribute Value
Availability	SUNWuto

SEE ALSO

`visual_io(7I)`

NAME

`utaction` - Sun Ray appliance connect / disconnect action

SYNOPSIS

/opt/SUNWut/lib/utaction [`-c` *ccmd*] [`-d` *dcmd*] [`-D` *display*] [`-i`] [`-t` *sec*]

DESCRIPTION

The `utaction` program provides a way to execute commands when a Sun Ray appliance session is connected or disconnected. The *ccmd* is invoked using `sh`(1) whenever the session is connected to an appliance. Similarly, the *dcmd* is invoked using `sh`(1) whenever the session is disconnected from an appliance. Normally, action is not taken on the initial state of the session (when `utaction` is first run) unless the -i option is used.

OPTIONS

The following options are supported.

`-c` *ccmd* — Run this command when the current session is connected to an appliance.

`-d` *dcmd* — Run this command when the current session is disconnected to an appliance.

`-D` *display* — This option will set the X display variable that is to be used in determining the Sun Ray enterprise appliance session. Otherwise the DISPLAY environment variable is used.

`-i` — Run the connect or disconnect command immediately, whichever is appropriate.

`-t` *sec* — This option specifies a time-delay in seconds for the actions. In that case, the *ccmd* or *dcmd* will not be invoked unless the session remains in the connected or disconnected state, respectively, for at least *sec* seconds.

EXAMPLES

EXAMPLE 1 This command invokes the CDE screen lock whenever the session is disconnected

```
% utaction -d '/usr/dt/bin/dtaction LockDisplay' &
```

ATTRIBUTES

See `attributes(5)` for descriptions of the following attributes:

Attribute Type	Attribute Value
Availability	SUNWuto

NOTES

The *ccmd* and *dcmd* are each only one argument to `utaction`. Quotes should be used if a command contains multiple words.

NAME

`utadem` - Sun Ray audio driver emulator.

SYNOPSIS

`/dev/utadem`

DESCRIPTION

`utadem` provides a generic virtual audio interface to Sun Ray appliances. The actual interface to the appliance is through a daemon that is session-aware. The daemon connects to `utadem` through a master port and is responsible for creating the slave device nodes which connect to normal audio applications.

API

Applications that normally open `/dev/audio` may use `utadem` as long as they have some way of selecting the audio device, such as through the `-d` *device* switch, or the AUDIODEV environment variable. The exact capabilities of the audio device emulated depend on the daemon. Compliance to the standard `audio`(7I) interface is handled in the following manner:

Audio Data Formats	The data formats supported depend on the daemon. Please refer to the daemon documentation for its capabilities.
Audio Ports	Input and output audio ports are directly dependent on the Sun Ray appliance and not on the daemon. The daemon is capable of discovering the type and quantity of input ports available and report them in the `record.avail_ports` and `play.avail_ports` fields of the `audio_info` structure. Although the ports can be controlled directly, the actual audio output is generally a mix of multiple services, so the `play.gain` setting is the contribution of this audio device to the total experience. Since recording is exclusive of a single service, the `record.gain` and `record.balance` controls directly affect the hardware gain.
Sample Granularity	Since the `utadem` driver is working through a daemon which transfers the audio data over an interconnect, larger granularities and jitter in the reporting of sample counts is possible. At any given time, the reported input and output sample counts will vary from the actual sample count by no more than the size of the buffers it is transferring. Programs should not rely upon the absolute accuracy of the `play.samples` and `record.samples` fields of the `audio_info` structure.

Audio Status Change Notification	As described in audio(7I), it is possible to request asynchronous notification of state changes in an audio device.

ERRORS

utadem errors are defined in the audio(7I) man pages. If the daemon has exited, further audio operations are no longer possible on the slave ports. Audio programs must exit in order to clear this error. New opens will return ENODEV. Data writes and ioctl operations will return ENXIO. Data reads will complete normally and then return end-of-file.

FILES

The following file is used:

- /dev/utadem

  Master port for daemons.

The logical device name of the slave port depends on the daemon.

ATTRIBUTES

See attributes(5) for descriptions of the following attributes:

Attribute Type	Attribute Value
Availability	SUNWutu
MT-Level	Safe

SEE ALSO

utaudio(1), ioctl(2), attributes(5), audio(7I), streamio(7I)

NAME

`utadm` - Sun Ray network and DHCP configuration utility.

SYNOPSIS

/opt/SUNWut/sbin/utadm `-a` *interface-name* [`-a` *interface-name*]...

/opt/SUNWut/sbin/utadm `-c`

/opt/SUNWut/sbin/utadm `-d` *interface-name* [`-d` *interface-name*]...

/opt/SUNWut/sbin/utadm `-f`

/opt/SUNWut/sbin/utadm `-n`

/opt/SUNWut/sbin/utadm `-p`

/opt/SUNWut/sbin/utadm `-r`

DESCRIPTION

The `utadm` command manages the private network and DHCP configuration for the Sun Ray interconnect. It configures the name lookup, host, network, netmask, and DHCP database files so that Sun Ray appliances can be connected to a central server host over one or more private subnets. One of the following option flags must be specified: `-a`, `-c`, `-d`, `-f`, `-n`, `-p`, or `-r`. The command is run with superuser privileges.

OPTIONS

The following options are supported.

`-a` Configure the network interface specified by *interface-name* as an Sun Ray subnetwork. In the default case, an available private subnetwork address is selected from the range 192.168.128.0 to 192.168.254.0. If the subnet selected is 192.168.*N*.0, entries for the hosts, networks, and netmasks files are generated using the *hostname* of the server and *interface-name*:

File	Entry
`/etc/hosts`	`192.168.N.1` *hostname-interface-name*
`/etc/networks`	`SunRay-`*interface-name* `192.168.`*N*`.0 SunRay`
`/etc/netmasks`	`192.168.`*N*`.0 255.255.255.0`

Once these entries are established, the network interface is activated as hostname-interface-name using `ifconfig(1M)`. If the interface is already up and configured, the user will be given the option to bypass configuration of the network interface and only configure DHCP on the interface. This allows configuration of a Sun Ray interconnect on the primary interface of the server. IP addresses on the Sun Ray subnets are managed using the DHCP protocol, which requires the addition of several macro entries to the `dhcptab(4)` table to control parameters on Sun Ray subnets. It is also possible to bypass DHCP configuration by entering 0 as the first unit address when prompted. The `pntadm(1M)` command is also used to create the pool of available IP addresses for assignment to Sun Ray appliances. Once the interface is configured and activated, `utfwadm(1M)` is invoked to add the current version of the firmware to the DHCP macros for the new network. The user is prompted for approval of all the default options, and may change them as desired.

The `-a` option implies the `-c` option if the initial configuration has not yet been performed.

`-c` Initialize the basic configuration files for a Sun Ray interconnect without setting up any subnetworks. This involves making sure that the network database files and framework for DHCP exist, and setting the `/etc/nsswitch.conf` file so that network information for the local Sun Ray subnets is obtained from local files.

`-d` Delete the network interface specified by *interface-name* from the list of configured Sun Ray subnetworks. The specified interface must have been previously configured using the `-a` option.

`-f` Takes this server offline, which prevents the creation of new sessions on this server when it is within a failover group. Existing sessions will not be killed, but load balancing will not select this server for new sessions.

`-n` Bring this server back online. This restores normal operation of the server and allows new sessions to be created on this server.

`-p` Print the current Sun Ray interconnect configuration, showing for each interface the hostname, network, netmask, and number of IP addresses out of an available pool of addresses assigned to Sun Ray appliances by DHCP.

`-r` Unconfigure all active Sun Ray interfaces and remove all Sun Ray entries from the configuration databases.

EXAMPLES

EXAMPLE 1 The following example configures the Sun Ray private network on `hme1`

```
# /opt/SUNWut/sbin/utadm -a hme1
```

FILES

The following files are used:

- `/etc/nsswitch.conf`
  Name service switch configuration file.
- /var/dhcp/dhcptab
  File or NIS+ table
- `/etc/default/dhcp`
  DHCP service configuration file.
- `/etc/inet/hosts`
  File or NIS+ table
- /etc/inet/networks
  File or NIS+ table
- `/etc/inet/netmasks`
  File or NIS+ table
- /etc/hostname.*
  Hostname for each interface.

ATTRIBUTES

See `attributes(5)` for descriptions of the following attributes:

Attribute Type	Attribute Value
Availability	SUNWuto

SEE ALSO

`ifconfig(1M)`, `dhtadm(1M)`, `pntadm(1M)`, `dhcpconfig(1M)`, `syslogd(1M)`, `syslog(3)`, `dhcp(4)`, `dhcp_network(4)`, `dhcptab(4)`, `nsswitch.conf(4)`, `hosts(4)`, `networks(4)`, `netmasks(4)`, `syslog.conf(4)`, `attributes(5)`, `utfwadm(1M)`

Alexander, S., and Droms, R., DHCP Options and BOOTP Vendor Extensions, RFC 1533, Lachman Technology, Inc., Bucknell University, October 1993.

Droms, R., Dynamic Host Configuration Protocol, RFC 1541,Bucknell University, October 1993.

NAME

`utadmin.conf` - Sun Ray server administration configuration file.

SYNOPSIS

`/etc/opt/SUNWut/utadmin.conf`

DESCRIPTION

The `utadmin.conf` file is a standard Java properties file that contains configuration parameters for the Sun Ray server administration database. It is usually installed by `utinstall`(1M) and configured by `utconfig`(1M).

The `admin.defaultlocale` parameter (see below) is the only parameter that should be changed once the Sun Ray server is configured and in use. All other parameters are reserved.

PROPERTIES

The supported configuration parameters are listed below. For each one, the name, description, and an example are given.

Name	Description
`admin.defaultlocale`	The default locale for the Web-based Administration Tool. Supported values are "en_US" (US English), "fr" (French), "ja" (Japanese) and "zh" (Simplified Chinese). Example: en_US
`admin.dstatus.dbfile`	The name of the NDBM data files where the desktop status is stored. Example: `/var/opt/SUNWut/ndbm/dstatus.dir` `/var/opt/SUNWut/ndbm/dstatus.pag`
`admin.http.cfile`	Configuration file for the Sun Ray Administration Server. Default is the `/etc/http/utadmin.http.conf` file.
`admin.http.port`	The webserver port used by the Administration Tool. Default is 1660.
`admin.server.name`	The name of the server where the administration database LDAP server process is running. This is usually the host name of the Sun Ray server. Example: `yoyodyne`
`admin.server.port`	The administration database LDAP server port. This is usually port 389. Example: 389

admin.ssl.enable	Secure connection between browser and server using SSL. Value: yes - SSL is enabled no - SSL is disabled
admin.subtree	The subtree within the LDAP hierarchy where Sun Ray administration data for this server resides. This is an entry under the UT root entry that was specified by utconfig. Example: utname=yoyodyne,o=v1,o=utdata
admin.user.name	The LDAP user that the administration clients should bind as to perform privileged operations. Example: cn=utadmin,utname=yoyodyne,o=v1,o=utdata
admin.ustatus.dbfile	The name of the NDBM data files where the user status is stored. Example: /var/opt/SUNWut/ndbm/ustatus.dir /var/opt/SUNWut/ndbm/ustatus.pag

EXAMPLES

EXAMPLE 1 Configuration parameters for the LDAP and NDBM databases:

```
admin.server.name      = sray-139
admin.server.port      = 389
admin.user.name        = cn=utadmin,utname=sray-139,o=v1,o=utdata
admin.subtree          = utname=sray-139,o=v1,o=utdata
admin.defaultlocale    = en_US
admin.dstatus.dbfile   = /var/opt/SUNWut/ndbm/dstatus
admin.ustatus.dbfile   = /var/opt/SUNWut/ndbm/ustatus
admin.http.cfile       = /etc/http/utadmin.httpd.conf
admin.http.port        = 1660
admin.ssl.enable       = no
```

ATTRIBUTES

See attributes(5) for descriptions of the following attributes:

Attribute Type	Attribute Value
Availability	SUNWutr

SEE ALSO

utinstall(1M), utconfig(1M), utuser(1M), utdesktop(1M), *Sun Ray Server Software 1.3 Administrator's Guide*

NAME

`utaudio` - Sun Ray audio services connection utility.

SYNOPSIS

/opt/SUNWut/bin/utaudio

csh

setenv AUDIODEV ‘utaudio‘

ksh

export AUDIODEV=‘utaudio‘

sh

AUDIODEV=‘utaudio‘;export AUDIODEV

DESCRIPTION

`utaudio` enables standard Solaris audio services using the `utadem`(7D) audio device emulator driver. After connecting to a Sun Ray session, `utadem`(7D) creates a new audio device for which `utaudio` creates device files in the `/tmp/SUNWut/dev` directory. `utaudio` then echoes the root device name to standard output, setting the AUDIODEV environment variable. Standard audio applications can then open the new audio pseudo-device and perform audio play and record operations.

OPTIONS

There are no options for `utaudio`.

API

Applications that use the `/dev/audio` interface may open the device indicated by the AUDIODEV environment variable and use the AUDIO_GETDEV `ioctl` to determine which audio device is being used. The `utaudio` driver returns the string "SUNW,CS4231" in the name field of the `audio_device` structure to indicate compatibility with other Ultra platforms. The version field contains "a” and the config field contains "pseudo."

The AUDIO_SETINFO `ioctl` controls device configuration parameters. When an application modifies the `record.buffer_size` field using the AUDIO_SETINFO `ioctl`, the daemon will constrain it to be non-zero and up to a maximum of 8180 bytes.

Audio Data Formats

The `utaudio` daemon supports u-law and A-law with 8-bit precision or 16-bit linear PCM at any sample rate from 8000 Hz to 48 kHz for one or two channels. The Sun Ray standard sampling rate is 48 kHz as this yields the best quality. The input and output data formats for playing and recording do not have to match. Some input devices do not provide 2-channel capture, but two channels will be reproduced by duplication in the case where two channels are requested and the device supports only one.

Audio Ports

The record.avail_ports and play.avail_ports fields of the audio_info structure report the available input and output ports for the currently connected Sun Ray appliance. Only AUDIO_MICROPHONE and AUDIO_LINE_IN are supported and most devices will have both inputs. The Sun Ray audio model supports individual volume controls for the two, so it is possible that the volume setting will change with input.

For output, AUDIO_LINE_OUT is always selected and does not have variable gain. AUDIO_SPEAKER and AUDIO_HEADPHONE are supported and they share a level control. In general, comfortable settings for the speaker will also be comfortable for headphone use. Either or both outputs can be selected simultaneously. The Sun Ray specification supports a third, automatic switching mode that is accessed by deselecting both speaker and headphone or by selecting only line out. The utsettings(1) command may also be used to control the device's outputs. In automatic mode, the settings track the physical connection of the headphone.

FILES

The following files are used:

- /tmp/SUNWut/dev/utaudio/*n*

  Numbered audio data pseudo-device file nodes.

- /tmp/SUNWut/dev/utaudio/*n*ctl

  Matching numbered control pseudo-device file nodes.

ENVIRONMENT VARIABLES

utaudio requires the DISPLAY environment variable contain an X11(7) display for which the user's session has access. This is set-up automatically in the Sun Ray environment.

An alternate driver emulator or different unit number can be specified in the UT_ADEM environment variable.

The results of utaudio should be placed in the AUDIODEV environment variable.

EXIT STATUS

The following exit values are returned:

0	Normal completion -- daemon back grounded
1	Either the X11 server, or the session could not be contacted, or there was a problem creating the new pseudo audio device.

ATTRIBUTES

See `attributes(5)` for descriptions of the following attributes:

Attribute Type	Attribute Value
Availability	SUNWuto

SEE ALSO

`utsettings(1)`, `X11(7)`, `utadem(7D)`, `audio(7I)`, `steamio(7I)`, `ioctl(2)`, `priocntl(2)`, `attributes(5)`, `environ(5)`

NOTES

The `audio`(7I) interface does not have an interface for dynamically changing audio devices such as that offered by the Sun Ray software. It is not possible to track the movement of sessions or changes in audio hardware using this device interface. The `utaudio` daemon makes a best-effort attempt to report changes in device control ability and to make the device appear as flexible as possible, matching that ability to the actual Sun Ray hardware being used.

If a session is disconnected, audio output continues as if there was an actual hardware connection, even though no samples are actually being transmitted or played. Conversely, audio input stops when there is no connected device.

NAME

`utauthd` - Sun Ray appliance authentication daemon.

SYNOPSIS

/opt/SUNWut/lib/utauthd `-b | -e [-d]`

DESCRIPTION

The `utauthd` daemon is responsible for authentication and access control for the Sun Ray appliances attached to a server. This command should not be executed directly. It is invoked by a system startup script.

OPTIONS

The following options are supported.

`-b` Begin execution of the daemon.

`-d` UNSUPPORTED. Enable debug log messages. These messages can reveal authentication secrets. This option should not be used in a production environment.

`-e` End execution of the daemon.

Without arguments, the default is `-b`.

FILES

The following files are used by the daemon:

- `/etc/init.d/utsvc`

  This is the system startup script that invokes the daemon. `/opt/SUNWut/utsessiond`, the Session Manager, performs the actual session switching function.

- `/etc/opt/SUNWut/auth.props`

  The Authentication Manager's configuration file.

- `/etc/opt/SUNWut/policy/utpolicy`

  This file determines what policy is used by the Sun Ray server.

The authentication manger is normally started by running `utsvc` with the start or restart argument. The start argument starts both the session manager and the authentication manager, so all of the sessions are lost. The restart argument only starts the authentication manager, so all the sessions are continued.

ATTRIBUTES

See `attributes(5)` for descriptions of the following attributes:

Attribute Type	Attribute Value
Availability	SUNWuto

SEE ALSO

`auth.props(4)`, `utpolicy(1M)`

NAME

`utcapture` - Capture packet information from the Authentication Manager.

SYNOPSIS

/opt/SUNWut/sbin/utcapture `-h`

/opt/SUNWut/sbin/utcapture [`-r`] [`-s` *server*] [*desktopID1 desktopID2* `...`]

/opt/SUNWut/sbin/utcapture `-i` *filename*

DESCRIPTION

The `utcapture` command connects to the Authentication Manager and monitors packets sent and packets dropped between the Sun Ray server and the Sun Ray appliances.

utcapture writes the captured information to `stdout` in the following format:

```
TERMINALID TIMESTAMP TOTAL PACKET TOTAL LOSS BYTES SENT PERCENT LOSS
```

OPTIONS

The following options are supported.

`-h` Help for using the command.

`-i` *filename* Use an input file to search for Sun Ray appliances that experienced dropped packets. A file is created using `utcapture`:

`/opt/SUNWut/sbin/utcapture -r > /tmp/`*filename*

The process is allowed to run for several minutes or hours. The utcapture command is used again:

`/opt/SUNWut/sbin/utcapture -i /tmp/`*filename*

The output is only the appliances that experienced dropped packets.

`-r` Write captured data to stdout every 15 seconds in raw, continuous format.

`-s` *server* Specify the Sun Ray *server* from which to capture data. If outside of the domain of the host running `utcapture`, the Sun Ray server hostname must be fully qualified. By default, the server monitored is the host running `utcapture`.

When no option is specified, `utcapture` writes to `stdout` if there is any change in packet loss for any Sun Ray appliance at 15 second intervals.

OPERANDS

The following operands are supported:

desktopID Capture data for the specified Sun Ray appliances only. Appliances are specified by their Ethernet address (*desktopID*) separated by spaces. By default, data for all appliances is displayed.

EXAMPLES

EXAMPLE 1 This command captures data from the Authentication Manager running on `localhost` every 15 seconds and then writes it to `stdout` if there is any change in packet loss for any Sun Ray appliance.

```
% utcapture
```

EXAMPLE 2 This command captures data from the Authentication Manager running on `localhost` every 15 seconds and then writes it to `stdout` regardless if there is any change in packet loss.

```
% utcapture -r
```

EXAMPLE 3 This command captures data from the Authentication Manager running on `netraj118.eng` every 15 seconds and then writes it to `stdout` if there is any change in packet loss for appliances with the Ethernet address of `080020a893cb` or `080020b34231`.

```
% utcapture -r netraj118.eng 080020a893cb 080020b34231
```

EXIT STATUS

The following exit values are returned:

`0` Successful completion

`1` Error

ATTRIBUTES

See `attributes(5)` for descriptions of the following attributes:

Attribute Type	Attribute Value
Availability	SUNWuta

SEE ALSO

`utauthd(1M)`, `utdesktop(1M)`

NOTES

`utcapture` does not report packet information for Sun Ray appliances using firmware versions of 1.1 or less.

NAME

`utcard` - Sun Ray server smart card configuration utility

SYNOPSIS

/opt/SUNWut/sbin/utcard `-a` *filename*

/opt/SUNWut/sbin/utcard `-d` *name,version*

/opt/SUNWut/sbin/utcard `-h`

/opt/SUNWut/sbin/utcard `-l`

/opt/SUNWut/sbin/utcard `-p` *name,version*

/opt/SUNWut/sbin/utcard `-r` *name,version,new-position*

/opt/SUNWut/sbin/utcard `-u`

DESCRIPTION

The `utcard` command allows configuration of different types of smartcards in the Sun Ray administration database.

The administrator must first place a configuration file for a specific smartcard in the `/etc/opt/SUNWut/smartcard` directory. This file must have a `.cfg` extension. The smartcard definition in the `.cfg` file is added to the LDAP datastore by using the `-a` option. When a smartcard definition is added, it is automatically assigned the last position in the probe order.To modify the probe order, use the `-r` option.

OPTIONS

The following options are supported.

`-a` *filename*	Add the card specified within *filename* that is in the `/etc/opt/SUNWut/smartcard` directory
`-d`	Delete the card specified with *name, version*.
`-h`	Show usage information
`-l`	List all configured cards
`-p`	Show the standard properties for the card specified with *name, version*.
`-r`	Reorder the card specified with *name, version, to new-position*.
`-u`	List unconfigured cards available for configuration as determined by the `.cfg` files in `/etc/opt/SUNWut/smartcard`

USAGE

Use this command only on a Sun Ray server that has been configured for administration by the `utconfig` command.

ATTRIBUTES

See attributes(5) for descriptions of the following attributes:

Attribute Type	Attribute Value
Availability	SUNWuto

SEE ALSO

utconfig(1M)

NAME

`utconfig` - Sun Ray server software configuration utility.

SYNOPSIS

/opt/SUNWut/sbin/utconfig [`-u`]

DESCRIPTION

The `utconfig` command performs initial configuration of Sun Ray server and supporting administration framework software. Before taking any actions the command prompts the user for configuration parameters for each of the supporting software packages. The command must be run with superuser privileges.

OPTIONS

The following option is supported.

`-u` Unconfigure the Sun Ray server and administration software returning the mode of operation back to the default zero administration mode.

ATTRIBUTES

See `attributes(5)` for descriptions of the following attributes:

Attribute Type	Attribute Value
Availability	SUNWuto

SEE ALSO

`patchadd(1M)`, `pkgadd(1M)`, `pkgrm(1M)`, `admin(4)`, `utinstall(1M)`

NAME

`utdesktop` - Sun Ray appliance administration utility.

SYNOPSIS

/opt/SUNWut/sbin/utdesktop `-a` *"desktopID,location,otherinfo"*

/opt/SUNWut/sbin/utdesktop `-a -f` *filename*

/opt/SUNWut/sbin/utdesktop `-d` *desktopID*

/opt/SUNWut/sbin/utdesktop `-d -f` *filename*

/opt/SUNWut/sbin/utdesktop `-e` *"desktopID,location,otherinfo"*

/opt/SUNWut/sbin/utdesktop `-e -f` *filename*

/opt/SUNWut/sbin/utdesktop `-h`

/opt/SUNWut/sbin/utdesktop `-l [-c | -g | -w [-t` *timeout*`]]`

/opt/SUNWut/sbin/utdesktop `-L {-c | -w [-t` *timeout*`]}`

/opt/SUNWut/sbin/utdesktop `-l -i` *substring*

/opt/SUNWut/sbin/utdesktop `-o`

/opt/SUNWut/sbin/utdesktop `-p` *desktopID*

DESCRIPTION

The `utdesktop` command allows the user to manage Sun Ray appliances connected to the Sun Ray server the command is run on. The information that `utdesktop` displays and allows the user to add, edit, or delete is stored in the Sun Ray administration database. Other information is obtained from the Sun Ray authentication manager.

`utdesktop` operations that only display information may be run by any user. Operations that add, edit, or delete data must be run by the superuser.

OPTIONS

The following options are supported.

`-a` *desktopID,location,otherinfo*	Add appliance with the specified desktop-ID, location, and other information properties. Note that the 3 comma-delimited values should be enclosed within quotes. You must be root to use this option.
`-a -f` *filename*	Batch add multiple appliances using input from the specified filename. The format of each line in the input file is: desktop-ID,location,other-info. You must be root to use this option.
`-d` *desktopID*	Delete the appliance with the specified desktop-ID. You must be root to use this option.

`-d -f` *filename*	Batch delete multiple appliances using input form the specified filename. The format of each line in the input file is: desktop-ID. You may use the output of the `-o` option to feed this option as all arguments after the first comma are ignored. You must be root to use this option.
`-e` *desktopID,location,other info*	Edit properties for the specified appliance by changing the location and other information properties to the specified values. Note that the 3 comma-delimited values should be enclosed within quotes. You must be root to use this option.
`-e -f` *filename*	Batch edit properties for multiple appliances using input from the specified filename. The format of each line in the input file is: desktop-id,location,other-info You must be root to use this option.
`-h`	Show usage information (this message).
`-l`	List all appliances.
`-l -c`	List all appliances that are currently connected.
`-L -c`	List all appliances that are currently connected (long format).
`-l -g`	List all currently connected appliances and the servers they are connected to.
`-l -w [-t` *timeout*]	List all appliances waiting for a session during the set *timeout* (short format). The default value of the timeout is 60 seconds.
`-L -w [-t` *timeout*]	List all appliances waiting for a session during the set *timeout* (long format). The default value of the timeout is 60 seconds.
`-l -i` *substring*	List all appliances with desktop IDs that contain the specified substring.
`-o`	Dump appliance list in comma-delimited format. The format of each line output by this option is: desktop-id,location,other-info
`-p`	Show desktop properties for the appliance with the specified ID.

EXAMPLES

EXAMPLE 1 This command clears the location and the other information properties for appliance 080020a85112:

```
# utdesktop -a "080020a85112,,"
```

EXAMPLE 2 This command changes the location and the other information properties for appliance 080020a85112 to "SFO12-2103" and "John's Office", respectively:

```
# utdesktop -e "080020a85112,SFO12-2103,John's Office"
```

EXAMPLE 3 This command edits the properties of multiple appliances using input from the file `/tmp/desktops`:

```
# utdesktop -e -f /tmp/desktops
```

EXAMPLE 4 This command displays all appliances that contain "a851" in their desktop IDs:

```
% utdesktop -l -i a851
```

EXAMPLE 5 This command lists all appliances in an error state without sessions within the default timeout:

```
% utdesktop -l -w
```

EXAMPLE 6 For a busy or slow network, this command lists (in long format) all appliances in an error state without sessions for at least five minutes:

```
% utdesktop -L -w -t 300
```

EXAMPLE 7 This command displays the current properties for appliance 080020a85112:

```
% utdesktop -p 080020a85112
```

FILES

The following file is used:

`/etc/opt/SUNWut/utadmin.conf`

ATTRIBUTES

See `attributes(5)` for descriptions of the following attributes:

Attribute Type	Attribute Value
Availability	SUNWuta

SEE ALSO

`utuser(1M)`, `utadmin.conf(4)`, the *Sun Ray Server Software 1.3 Administrator's Guide*

NOTES

The `-G` option has been deprecated. Use the `-l -g` option pair instead.

NAME

`utdetach` - Detach the current session from the Sun Ray appliance.

SYNOPSIS

/opt/SUNWut/bin/utdetach

DESCRIPTION

The `utdetach` command disconnects the current session from its respective Sun Ray appliance. The session is not destroyed, but rather put into a disconnected state. The session can be accessed if the same user token is presented to the Sun Ray server.

This command is primarily executed by users of the non-smart card mobility feature so as to disconnect their "mobile" sessions.

The Sun Ray server starts an instance of `utslaunch` (1M) for each session whenever a user logs into a Sun Ray appliance via `dtlogin`. This makes the `utdetach` command available to users as a hotkey sequence. The default hotkey sequence is Shift + Pause and can be configured in the `utslaunch.properties` file.

OPTIONS

There are no options for `utdetach`.

EXAMPLES

EXAMPLE 1 This command disconnects the current session from the appliance the user is currently using.

```
% utdetach
```

FILES

The following files are used:

- `/etc/opt/SUNWut/utslaunch_defaults.properties`
  site-wide defaults
- `~/.utslaunch.properties`
  user's defaults
- `/etc/opt/SUNWut/utslaunch_mandatory.properties`
  site-wide mandatory defaults

EXIT STATUS

The following exit values are returned:

`0`	Success
`1`	Failure

ATTRIBUTES

See attributes(5) for descriptions of the following attributes:

Attribute Types	Attribute Values
Availability	SUNWuto
Stability Level	Evolving

SEE ALSO

utslaunch(1M), utslaunch.properties(4)

NAME

`utdevmgrd` - Sun Ray device manager daemon.

SYNOPSIS

/opt/SUNWut/lib/utdevmgrd [`-a` *authlist*] [`-c` *authfile*] [`-d`] [`-k` *authprops*] [`-o` *optroot*] [`-p` *port*] [`-r`] [`-s` *sigfile*] [`-t`]

DESCRIPTION

The `utdevmgrd` daemon is responsible for brokering devices that are attached to Sun Ray appliances on the interconnect fabric for the purpose of remotely accessing the devices for various services. It is also responsible for approving services, keeping an inventory of devices and their controlling services, and locating devices on the interconnect.

If either the `-a` or the `-c` option is specified, the device manager daemon operates exclusively in call-back mode. In this mode, the device manager only communicates to authentication managers that are explicitly enabled by *authlist* or *authfile* and that have requested a call-back. The call-back feature provides a mechanism by which the device manager and the authentication manager establish each other's identity.

The *optroot* directory (default `/tmp/SUNWut`) is shared with other Sun Ray server components. Primarily it provides the location for a Solaris compatible device tree for each Sun Ray appliance's devices in the `sessions` and `units` subdirectories.

The `units` subdirectory contains a directory for each appliance on the interconnect named by the appliance's serial number. Within an appliance's directory, there are the familiar `dev` and `devices` directories that list logical names for devices and geographically hierarchical names for devices.

The `sessions` directory contains symbolic links into the `devices` directory that indicate which sessions are connected to which Sun Ray appliances. The symbolic links are named after the X-Windows server display corresponding to a user's session by display number only (in other words, after removing the server name, which is always a name local to the current host, and the screen numbers). The user's `DISPLAY` environment variable can then be used to find the devices on the 'current' appliance. The user's `UTDEVROOT` environment variable achieves this, and can be used to find devices that are 'currently' accessible. The *optroot* directory also includes the named pipe with which the device manager communicates to device driver services and the `session_info` directory, which contains user information important to internal workings of the device manager.

The device manager works within a Sun Ray server group environment, which enables rapid switching to other servers and user load distribution. In order for device managers on each server in a group to communicate, the device manager must gain access to the group signature file. If the signature does not match the one used by other device managers in the group, then grouping will fail and not all devices on all appliances on the interconnect will be available on the server, including devices on some appliances being used by users on the server.

Normally, the device manager finds the group signature file by looking into the authentication manager's configuration file (`/etc/opt/SUNWut/auth.props`),

but this can be changed by using the -s and -k options. If -s is specified, then *sigfile* is read and used as the group signature. If -k is specified, then the *authprops* file is scanned for the gmSignatureFile key and the listed file is used for the group signature.

Error messages from utdevmgrd are logged using syslog(3), with a facility value of LOG_DAEMON.

OPTIONS

The following options are supported.

-a *authlist*	Add the host and port pairs specified in *authlist* to the list of allowed authentication managers. The format of *authlist* is a comma separated list of *hostname*:*port* pairs.
-c *authfile*	Add the host and port pairs specified in the ASCII file *authfile* to the list of allowed authentication managers. The file contains a list of authentication manager specifications, one per line. The specifications take the form of hostname followed by port number, separated by white space. Blank lines and any line whose first printable character is ‘‘#’’ are ignored.
-d	Enable debugging output.
-k *authprops*	Set the location for the authentication manager’s configuration file to *authprops*. This file is used to find the group signature file in case the *sigfile* key was not specified. The default for this parameter is /etc/opt/SUNWut/auth.props. The key in this file that specifies the group signature is gmSignatureFile.
-o *optroot*	Set the device information root directory to *optroot*. This directory contains the service named pipe, and the units, sessions, and session_info directories. *optroot* is generally shared with other Sun Ray server components.
-p *port*	Set the device manager’s listen port to the specified port value. The device manager defaults to port 7011. This is the port by which device services and authentication managers contact the device manager.
-r	Automatically restart the device manager daemon if it exits. With this option, the device manager daemon creates two processes: a child that performs all the actual work and a parent monitoring process. The parent process will restart a child if the previous one exits. This enables existing services to re-attach to a new child device manager.
-s *sigfile*	Set the path of the group signature file to *sigfile*.
-t	Test mode. Relax checking for error returns for files that are root access. Could cause unpredictable results on an operational device manager in case of true failure.

FILES

The following files are used:

`/etc/opt/SUNWut/auth.permit`	The customary location of the *authfile* for a system.
`/tmp/SUNWut`	The customary location for temporary files used by Sun Ray enterprise server managers, designated by *optroot*.
`/tmp/SUNWut/.utdevmgr`	The named pipe used for communication between the device manager and device driver services.
`/tmp/SUNWut/units`	The directory containing device directories for each appliance. The directory names are after the appliances' serial numbers. Each directory contains a `dev` directory and a `devices` directory.
`/tmp/SUNWut/sessions`	The directory containing links to appliances in the `units` directory, named by X-Windows display number for each session. These links change as users move from one Sun Ray appliance to another.
`/tmp/SUNWut/session_info`	The directory containing information internal to the device manager for handling session mobility.
`/etc/opt/SUNWut/auth.props`	The customary location of the *authprops* file containing the authentication manager settings. The device manager looks for the gmSignatureFile key to extract the location of the group signature file.
`/etc/opt/SUNWut/gmSignature`	The customary location of the *sigfile* file containing the group signature.

ENVIRONMENT VARIABLES

The following environment variables are used:

`DISPLAY`	Use to get the default X-Windows display number from within the user's environment.
`UTDEVROOT`	Use to get the devices for the current session from within the user's environment.

ATTRIBUTES

See attributes(5) for descriptions of the following attributes:

Attribute Type	Attribute Value
Availability	SUNWuto

SEE ALSO

utauthd(1M), syslog(3), syslogd(1M), syslog.conf(4)

NAME

`utfwadm` - Sun Ray appliance firmware version management.

SYNOPSIS

/opt/SUNWut/sbin/utfwadm `-A` {`-a` | `-s` *enetSuffix* | `-e` *enetAddr*} `-n` *interf* [`-n` *interf*]... [`-f` *firmware*]

/opt/SUNWut/sbin/utfwadm `-D` {`-a` | `-s` *enetSuffix* | `-e` *enetAddr*} `-n` *interf* [`-n` *interf*]... [`-f` *firmware*]

/opt/SUNWut/sbin/utfwadm `-P`

/opt/SUNWut/sbin/utfwadm `-R`

DESCRIPTION

The `utfwadm` command manages firmware upgrades to Sun Ray appliances. The appliances are capable of loading firmware upgrades and programming new firmware into their flash PROM memory.

When an appliance is powered on, the firmware obtains an IP address and other configuration information using the DHCP protocol. Part of the configuration information is a firmware version identifier. If this identifier does not match the appliance's existing firmware, the appliance initiates an upgrade which replaces the current firmware with the new version.

The `utfwadm` command must be run when a new firmware version is installed to update the firmware version identifier and force the appliances to load the new version on their next power cycle. utfwadm allows firmware identifiers to be set on either a per-network or per-unit basis enabling firmware upgrades to be targeted at entire Sun Ray subnetworks or individual appliances.

It is possible to determine the firmware versions that are available and in use:

- In the `/tftpboot` directory are the firmware files. To verify the version of a firmware file, type:

  `/opt/SUNWut/sbin/lzd < /tftpboot/`*firmware-filename* `| what`

- To identify the version of firmware that a particular Sun Ray appliance is using, type:

  `/opt/SUNWut/sbin/utdesktop -p` *desktopID*

  Where *desktopID* is the full MAC address. This is displayed by pressing the three audio keys on the Sun Ray appliance keyboard simultaneously.

The Sun Ray subnetworks must have been previously set up using the `utadm(1M)` command. The `utfwadm` command is run under super-user privileges.

OPTIONS

The following options are supported.

`-A` This causes the given operation to be applied to all units attached to the given interfaces.

`-a`	Add the defined appliances to the list of units to be upgraded with a new firmware version. The following options determine which subset of the units should be upgraded. This option sets the firmware version identifier in the appropriate context. It also copies files from the firmware install directory into the boot directory, renaming them to contain their version strings.
`-D`	Remove the defined appliances from the list of units to be upgraded. This option causes the firmware version identifier to be unset.
`-e` *enetAddr*	This causes the operation to be applied to only the specified unit with Ethernet address given by *enetAddr*, where all six hex bytes of the address are specified.
`-f` *firmware*	This option gives the pathname for the firmware to be downloaded to the appliances. If *firmware* refers to a file, the hardware version is extracted from the version string within the file, and the file is copied to the `/tftpboot` directory to be downloaded only to that version of the hardware. If *firmware* refers to a directory, then all files named ‘‘Corona*’’ in the directory are copied to the `/tftpboot` directory with their version strings appended. If the `-f` option is not given, a default location is used.
`-n` *interf*	Apply the given operation to units connected to the Ethernet interface *interf*. Multiple interfaces may be given, or the special keyword `all`, which applies the operation to all configured Sun Ray interfaces.
`-P`	This variant of the command prints out the version to which each domain should be upgraded on the next power cycle. A domain may be either an interconnect subnet or individual appliance. If it is a subnet, then the `Intf` column lists the interface device. If it is an individual appliance, then its Ethernet address is given in the `Domain` column, and the `Intf` column contains the interface name.
`-R`	Remove the firmware files that were copied into the boot directory.
`-s` *enetSuffix*	This causes the operation to be applied to only the specified unit, where *enetSuffix* is given as the last three hex bytes of the Ethernet address. The hex prefix "0x080020" is added to the address.

The `-z` option is RESERVED for use by the Sun Ray server software and should not be used.

FILES

The following files are used:

- `/var/dhcp/dhcptab`

  File or NIS+ table

- `/tftpboot`

  Default location of firmware boot file

ATTRIBUTES

See `attributes(5)` for descriptions of the following attributes:

Attribute Type	Attribute Value
Availability	SUNWuto

SEE ALSO

`dhtadm(1M)`, `dhcpconfig(1M)`, `what(1)`, `dhcp(4)`, `dhcp_network(4)`, `dhcptab(4)`, `attributes(5)`, `utadm(1M)`, `utdesktop(1M)`

NAME

`utfwsync` - Synchronizes Sun Ray appliance firmware downloads.

SYNOPSIS

/opt/SUNWut/sbin/utfwsync [`-v`]

DESCRIPTION

The `utfwsync` command refreshes the firmware level on the Sun Ray appliances to what is available on the Sun Ray servers in an failover group and then forces all the Sun Ray appliances within the group to restart. This will cause each appliance to attempt to download the latest firmware offered by the primary Sun Ray server to which it is attached as it restarts, as described in `utfwadm(1M)`.

This command is intended for use after software upgrades or after new firmware has been installed on all hosts as part of a patch.

As the command executes access to user sessions is interrupted, but the sessions are not lost and will be returned after the command completes.

The command must be run with superuser privileges.

OPTIONS

The following option is supported.

`-v` Verbose mode. Additional messages regarding what is being done are written to `stdout`.

ATTRIBUTES

See `attributes(5)` for descriptions of the following attributes:

Attribute Type	Attribute Value
Availability	SUNWuta

SEE ALSO

`utgstatus(1M)`, `utauthd(1M)`, `utfwadm(1M)`, `utinstall(1M)`

NAME

`utglpolicy` - View or set policy for entire failover group.

SYNOPSIS

/opt/SUNWut/sbin/utglpolicy `-a` [`-g`] [`-k` *type*] [`-m`] [`-M`] [`-p`] [`-r` *type*] [`-s` *type*] [`-z` *type*]

/opt/SUNWut/sbin/utglpolicy `-a` [`-t` {`clear` | `add:`*terminalID* | `del:`*terminalID*}]

/opt/SUNWut/sbin/utglpolicy `-h`

/opt/SUNWut/sbin/utglpolicy

DESCRIPTION

The `utglpolicy` command returns or sets the `utpolicy` options which are used for the entire failover group. The `utglpolicy` command provides a command-line alternative to the Sun Ray Administration Tool which is normally used for setting group policies.

You must be `root` to use this command.

OPTIONS

The following options is supported.

Two categories of options are supported: Policy Setting and Card Reader Assignment

`-a` This option, followed by valid Policy Setting, or Card Reader Assignment arguments, applies these arguments to the active authentication policy for the failover group. This option is not valid by itself.

POLICY SETTING

The specified Policy Setting arguments completely replace the current active authentication policy. Only arguments that are specified become active. Policy Setting and Card Reader Assignment arguments can be specified together

`-g` Turn on session selection within a server group. Allows the user to select on which server the user's session is run.

`-k {card|pseudo|both}` Enables Controlled Access Mode (CAM) for the specified session type. Selecting card enables CAM for `card` sessions, `pseudo` for terminal sessions, and `both` enables CAM for all types of sessions. For functionality, at least one of the `-r`, `-s`, or `-z` options must be invoked with the same argument as the `-k` option. The `-k` option is not considered until the `utconfig` application has configured for Controlled Access Mode.

`-m`	Enable multihead session capability, allowing multiple terminals to act as display devices for a single user session.
`-M`	Enable non-smart card mobile sessions.
`-p`	This option changes the behavior of the self-registration application so that it does not require the Solaris name and password before registering a token. Note that the self-registration application only verifies the name and password. They are not stored.
`-r {card|pseudo|both}`	Specify the token types that must be registered in the administrative database in order to be granted access to a login screen. Policy looks up and uses token database entry
`-s {card|pseudo|both}`	Specify the token types that will be presented with a registration screen if they do not have an entry in the administrative database. Policy allows self-registration of tokens.
`-z {card|pseudo|both}`	Specify the token types that do not require an entry in the administrative database in order to be granted access to a login screen. Policy grants access to tokens without database entry.

CARD READER ASSIGNMENT

The Card Reader Assignment arguments are incremental in nature (a complete specification does not have to occur all at once). This means a card reader can be added today and another can be added next week. Both will then be active until explicitly deleted. Policy Setting and Card Reader Assignment arguments can be specified together

`-t clear`	Reset the list of Sun Ray appliances in dedicated card reader mode.
`-t add:`*terminalID*	Add a terminal (appliance) identification to the list of terminals being used as dedicated card readers. If a partial *terminalID* is specified, then the model will be assumed to be SunRayP1. If the *terminalID* is preceded by a backslash, then the *terminalID* will be used without any transformation.
`-t del:`*terminalID*	Remove a terminal (appliance) identification from the list of terminals being used as dedicated card readers.

Though similar to the `utpolicy` command, `utglpolicy` does not support the `-i {clear |soft}` and `-t list` options.

With the `-h` option, the `utglpolicy` command prints out the usage message.

With no options, the `utglpolicy` command prints out the policy in effect.

EXAMPLES

EXAMPLE 1 This command allows all appliances to be used with or without a smart card. Non-smart card mobile sessions are enabled.

```
# /opt/SUNWut/sbin/utglpolicy -a -M -z both
```

EXAMPLE 2 This command configures the policy so that all smart card users must be registered, however non-smart card users can still use the appliances. Controlled Access Mode is invoked for smart card users.

```
# /opt/SUNWut/sbin/utglpolicy -k card -r card -z pseudo
```

ATTRIBUTES

See `attributes(5)` for descriptions of the following attributes:

Attribute Type	Attribute Value
Availability	SUNWuta

SEE ALSO

`utpolicy(1M)`

NAME

`utgroupsig` - Sets the group signature for Sun Ray servers in a failover group.

SYNOPSIS

/opt/SUNWut/sbin/utgroupsig

DESCRIPTION

The `utgroupsig` command sets the failover group signature. It also sets the Sun Directory Services `rootpw` used by Sun Ray servers to a value based on the group signature.

The `utgroupsig` command prompts for the new signature twice. The group signature file is at least 8 bytes long and has similar content diversity characteristics as required by `passwd(1)`.

The signature is stored in clear in the location specified in the `auth.props` file with the `gmSignatureFile` property. The group signature file is created with owner `root` and mode 400 (read-only by root).

OPTIONS

There are no options for this command.

FILES

The following files are used:

- `/etc/opt/SUNWut/gmSignature`

  Sun Ray group signature default file.
- `/etc/opt/SUNWut/auth.props`

  Sun Ray authentication properties file.
- `/etc/opt/SUNWconn/ldap/current/dsserv.conf`

  Sun Directory Services Server configuration file.

EXIT STATUS

The following exit values are returned:

`0`	Success
`1`	Invalid UID. Run as `root`.
2	Unexpected failure. Signature file unchanged.

ATTRIBUTES

See `attributes(5)` for descriptions of the following attributes:

Attribute Type	Attribute Value
Availability	SUNWuta

SEE ALSO

`utrcmd(1M)`, `passwd(1M)`, `auth.props(4)`, `dsserv.conf(4)`

NAME

utgstatus - Display failover group status.

SYNOPSIS

/opt/SUNWut/sbin/utgstatus [-s *hostname*]

DESCRIPTION

The utgstatus command allows the user to view the failover group status information for the local server or for the named server. The information that the command displays is specific to that server at the time the command is run.

utgstatus displays information only and so can be run by any user.

OPTIONS

The following option is supported.

-s *hostname* Display all the failover group status information for the specified *hostname*.

EXAMPLES

EXAMPLE 1 This command displays the failover group status for the local Sun Ray server

```
% /opt/SUNWut/sbin/utgstatus
```

EXAMPLE 2 This command displays the failover group status for the server sun5:

```
% /opt/SUNWut/sbin/utgstatus -s sun5
```

Information returned from this command looks similar to the following (to view this correctly, make the terminal window very wide):

```
Server: sun5

                        Network/Netmask
        129.144.167.0/24 192.168.128.0/24 192.168.140.0/24 192.168.129.0/24
        ---------------- ---------------- ---------------- ----------------
sun5(T) 129.144.167.5(UP)192.168.128.2(SrU)192.168.140.1(SrU)
sun11(T)129.144.167.11(C)192.168.128.1(SrU)               192.168.129.1(NR)
sun55   129.144.167.55(C)                192.168.140.2(SrD)
```

Explanation of utgstatus information:

The Network/Netmask values are denoted in CIDR (Classless Inter Domain Routing) network address notation, where the initial value (129.144.167.0) is the network address itself and the '/24' suffix signifies the number of bits that are the network identifier of the address. The remaining 8 bits are for specific host addresses.

(T) Trusted — The trusted hosts are members of this failover group because they share the same group signature.

(UP) Up — The server sun5 is operational and communicating over the 129.144.167.5 interface.

(C) Connected — The interfaces at 129.144.167.11 and 129.144.167.55 are reachable by sun5.

(NR) Not Reachable — sun11 is not reachable by sun5 at address 192.168.129.1.

(SrU) / (SrD) Sun Ray Interconnect — The interfaces on the Sun Ray interconnect at 192.168.128.2 and 192.168.128.1 are up and available (SrU). The interface on the Sun Ray interconnect at 192.168.140.2 is not reachable (SrD).

ATTRIBUTES

See attributes(5) for descriptions of the following attributes:

Attribute Type	Attribute Value
Availability	SUNWuta

NAME

`utinstall` - Sun Ray server software installation, upgrade, and removal utility.

SYNOPSIS

/cdrom/cdrom0/utinstall [`-a` *admin-file*] [`-d` *media-dir*] [`-u`] [`-q`]

/opt/SUNWut/sbin/utinstall [`-a` *admin-file*] [`-d` *media-dir*] [`-u`] [`-q`]

DESCRIPTION

The `utinstall` command installs, upgrades, and removes Sun Ray server software. All software required to support the Sun Ray server is installed, including the administration framework, any patches required by the framework, and Solaris™ operating environment patches.

The `utinstall` command is run under superuser privileges and prompts the user before taking any action. Using the defaut administration file and media directory is recommended.

OPTIONS

The following options are supported.

`-a` *admin-file* — Instead of the default, use the *admin-file* as an installation administration file for `pkgadd` operations (see the `-a` option for `pkgadd(1M)`). The `admin_default` file, located at the installation media root, is used by default.

`-d` *media-dir* — Instead of the default, use *media-dir* as the installation media root directory.

`-q` — Quick install/remove. By appending this option, the `utinstall` command will automatically install or remove the software with no user interaction.

`-u` — Remove previously installed Sun Ray server software.

Without arguments, an interactive install or upgrade of the Sun Ray server software is performed.

FILES

The following files are used:

- `/cdrom/cdrom0/admin_default`
- `/opt/SUNWut/etc/admin_default`

These are the default installation administration files used by `pkgadd` operations.

ATTRIBUTES

See `attributes(5)` for descriptions of the following attributes:

Attribute Type	Attribute Value
Availability	SUNWuto

SEE ALSO

`patchadd(1M)`, `patchrm(1M)`, `pkgadd(1M)`, `pkgrm(1M)`, `admin(4)`

NAME

`utkiosk` - Sun Ray script to update kiosk configuration locally and within a failover group.

SYNOPSIS

/opt/SUNWut/sbin/utkiosk {`-e kiosk|-i kiosk`}

DESCRIPTION

The `utkiosk` script is used to import and export kiosk configuration information into the LDAP database. utkiosk also updates local configuration files in the `/var/opt/SUNWut/kiosk` directory and application working files in the `/var/opt/SUNWut/kiosk/config` directory. `utkiosk` is used primarily in failover groups.

OPTIONS

The following options are supported.

`-e kiosk` Only run on the secondary servers in a failover group or standalone server. If the LDAP server is running, synchronizes the secondary server's LDAP database with that of the primary server and the configuration is exported from LDAP replacing the local configuration file. The application working files are updated for the local configuration file.

`-i kiosk` Only run on the primary server in a failover group or standalone server. Updates the local kiosk configuration file and imports that configuration into the LDAP database if the LDAP server is running.

EXAMPLES

EXAMPLE 1 This command imports the configuration to LDAP and updates all local files.

```
# utkiosk -i kiosk
```

EXAMPLE 2 This command exports the configuration from LDAP and updates all local files

```
# utkiosk -e kiosk
```

FILES

The following files are used:

`/var/opt/SUNWut/kiosk/kiosk.conf`

All application working files in `/var/opt/SUNWut/kiosk/config`

EXIT STATUS

The following exit values are returned:

`0` Success

`1` Failure

ATTRIBUTES

See attributes(5) for descriptions of the following attributes:

Attribute Type	Attribute Value
Availability	SUNWuta

NAME

`utload` - Sun Ray appliance firmware download utility.

SYNOPSIS

/opt/SUNWut/sbin/utload [`-f` *firmware-file*] [`-h` *hostname*] [`-p` *port-number*] [`-s` *sessionID*] [`-w`]

DESCRIPTION

The `utload` command sends a request to a Sun Ray appliance to initiate a firmware download. As an option, the downloaded firmware can be written to the appliance's flash memory. The `utload` command is deprecated in the 1.3 release of the Sun Ray server software.

OPTIONS

The following options are supported.

`-f` *firmware-file* This option specifies the name of a file that must exist in the `/tftpboot` directory on the server. If the `-f` option is not specified, the default file "SunRayP1" is used.

`-h` *hostname* This option specifies the host running the Sun Ray authentication daemon (`utauthd`) that the appliance is connected to. The default is localhost.

`-p` *port-number* This option specifies the port number of the `utauthd` that manages the appliance. The default is port 7010.

`-s` *sessionID* This option specifies the appliance's current *sessionID*. The default value is derived from the current session if the command is invoked from a Sun Ray appliance. Please note that the *sessionID* should be kept secret. It is not advisable to use this command line option except in cases where session security is not important.

`-w` Write the downloaded firmware to flash memory.

If no options are specified, the command runs with the defaults.

FILES

The following files are used:

- `/tftpboot/SunRayP1`

  Default firmware file.
- `/etc/inetd.conf`

  Inet configuration file used to enable TFTP services.
- `/etc/opt/SUNWut/auth.props`

  Sun Ray authentication daemon configuration file.

ATTRIBUTES

See attributes(5) for descriptions of the following attributes:

Attribute Type	atTribute Value
Availability	SUNWuto

SEE ALSO

utadm(1M), utfwadm(1M)

NOTES

The utload command is deprecated in the Sun Ray server software 1.3. Future releases of the Sun Ray server software will not support the utload command.

NAME

`utmhadm` - Sun Ray appliance multihead group configuration utility.

SYNOPSIS

/opt/SUNWut/sbin/utmhadm [*groupname*]

/opt/SUNWut/sbin/utmhadm `-a` *groupname* `-g` *COLS*x*ROWS* `-p` *primaryCID* `-l` *CID1*, *CID2*,..., *CIDn*

/opt/SUNWut/sbin/utmhadm `-d` *groupname*

/opt/SUNWut/sbin/utmhadm `-e` [`-f` *filename*]

/opt/SUNWut/sbin/utmhadm `-o` [`-f` *filename*]

/opt/SUNWut/sbin/utmhadm `-h`

/opt/SUNWut/sbin/utmhadm

DESCRIPTION

The `utmhadm` command provides a way to administer Sun Ray server multihead multihead groups. The information that `utmhadm` displays and that is editable is stored in the Sun Ray administration database.

The `utmhadm` operations that only display information may be run by any user. Operations that change data must be run as superuser.

OPTIONS

The following options are supported.

`-a` *groupname*	Creates a new multihead group having identifier *groupname*. This name must be unique and not already exist on the system.
`-d` *groupname*	Removes the multihead group for the specified *groupname*.
`-e`	Populates the system multihead group database with input data of the format produced by `-o`, from standard input.
`-f` *filename*	Specify a *filename* for use with `-e` or `-o` instead of standard input or output.
`-g` *COLS*x*ROWS*	Specifies the geometry of the multihead group in the form *COLS*x*ROWS*. This number of columns and rows must not exceed the maximum number allowed and must match the number of appliances specified with `-l`. This option can only be used with `-a`.
`-h`	Prints the usage message.
`-l` *CID1*, *CID2*,..., *CIDn*	Specifies the appliance canonical identifiers when creating a group. A canonical identifier has the form IEEE802.*nnnnnnnnnnnn* or *nnnnnnnnnnnn* (the 12-digit hexadecimal MAC address of the appliance) and the list must be comma-separated. The identifiers must be specified in row-major order. The maximum number of appliances allowed is 16.

`-o` Dumps all system configured multihead group data, in comma-separated format, to standard output. Intended for subsequent use with `-e`.

`-p` *primaryCID* Identifies which in the list of canonical identifiers, specified with `-l`, is designated as the primary appliance within the group. The primary is repeated in the list specified by `-l`. This option can only be used with `-a`.

When no options are provided, `utmhadm` lists information about all multihead groups configured on the system.

EXAMPLES

EXAMPLE 1 This command list all appliances that are in the multihead group:

```
% /opt/SUNWut/sbin/utmhadm tera
```

Here is sample output:

```
Multihead Group    Geometry           CIDs
------------------ ------------------ --------------------------------
tera               geometry=2x1       IEEE802.080020b538dc (P)
                                      IEEE802.080020b56e2d
```

EXAMPLE 2 This command creates a terminal group having two terminals with the first one being the primary:

```
# /opt/SUNWut/sbin/utmhadm -a srgroupA -g 2x1 -p IEEE802.080020b0562f
-l IEEE802.080020b0562f,IEEE802.080020b64574
```

ATTRIBUTES

See `attributes(5)` for descriptions of the following attributes:

Attribute Type	Attribute Value
Availability	SUNWuta

SEE ALSO

`utxconfig(1)`, *Sun Ray Server Software 1.3 Administrator's Guide*

NAME

`utmhconfig` - Sun Ray mulithead GUI configuration utility.

SYNOPSIS

/opt/SUNWut/sbin/utmhconfig

DESCRIPTION

The `utmhconfig` utility allows the administrator to list, add, or delete multihead groups easily. The initial screen lists any existing multiheaded groups and allows the administrator to select those to delete. The utility can also be used to create a new group. To do this, the administrator starts the utility on the Sun Ray that is to become the "primary" of the group (it has the keyboard, mouse, and all the devices for the group). The administrator selects "Create New Group" and follows the instructions in the wizard to identify all of the terminals in the new multihead group. The administrator run the utmhconfig command as superuser and must have a recognized smart card available.

OPTIONS

There are no options for `utmhconfig`

ATTRIBUTES

See `attributes(5)` for descriptions of the following attributes:

Attribute Type	Attribute Value
Availability	SUNWuta

SEE ALSO

`utmhadm(1M)`, `utxconfig(1)`

NAME

`utmhscreen` - Sun Ray multihead GUI screen display tool.

SYNOPSIS

/opt/SUNWut/lib/utmhscreen

DESCRIPTION

The `utmhscreen` tool provides a window showing the respective displays location in the multihead group. The display showing the widow is highlighted in white, while the other displays are darkened. The window is located in the upper right corner of the display.

This tool is automatically launched for users during the X server startup process (session creation). If the X server is not running in a multihead environment, the tool immediately exits.

OPTIONS

There are no options for utmhscreen.

RESOURCES

The tool understands all of the core X Toolkit and Motif resource names and classes as well as:

enableAutoLaunch (class EnableAutoLaunch) — Specifies whether or not `utmhscreen` should be launched automatically during X session startup. The default is "true".

EXAMPLES

EXAMPLE 1 To disable automatic launching of `utmhscreen` for a user, set the following X resource in their `$HOME/.Xdefaults` file:

```
Utmhscreen*enableAutoLaunch: false
```

ATTRIBUTES

See `attributes(5)` for descriptions of the following attributes:

Attribute Types	Attribute Values
Availability	SUNWuta
Interface Stability	Evolving

SEE ALSO

`utmhadm(1M)`. `utmhconfig(1M)`, `utxconfig(1)`

NAME

`utparallel, utserial` - Sun Ray serial and parallel port device driver emulators.

SYNOPSIS

```
#include <sys/types.h>
#include <fcntl.h>
```

utserial

```
#include <sys/termios.h>
#include <termio.h>
```

utparallel

```
#include <sys/ecppio.h>
```

DESCRIPTION

`utserial` is a tty-style interface that provides a generic virtual interface to USB serial adaptors connected to the Sun Ray appliance.

`utparallel` is a parallel-style interface that provides a generic virtual interface to USB parallel adaptors connected to the Sun Ray appliance.

`utserial` and `utparallel` are each loadable `STREAMS` drivers.

EXTENDED DESCRIPTION

The actual interface to the appliance for each of these drivers is through the Sun Ray interconnect via either the `utseriald` daemon or the `utparalleld` daemon, each of which is session-aware. The daemons are connected to either `utserial` or `utparallel` through a master port and each is responsible for creating the slave device nodes through which normal applications will connect.

API

Applications open a device file created by either `utseriald` or `utparalleld`. Device files created by `utseriald` comply to the `termio`(7I) interface and device files created by `utparalleld` comply to the `ecpp`(7D) interface. Hardware limitations in USB adaptors might prevent compliance with these interfaces.

FILES

The following files are used:

- `/dev/utserial`

  Master port for `utserial`
- `/dev/utparallel`

  Master port for `utparallel`

ATTRIBUTES

See `attributes`(5) for descriptions of the following attributes:

Attribute Type	Attribute Value
Availability	SUNWutu
MT-Level	Safe

SEE ALSO utseriald(1M), utparalleld(1M), termio(7I), ecpp(7D)

NAME

utparalleld - Sun Ray printer service daemon.

SYNOPSIS

/opt/SUNWut/lib/utparalleld [-D *debug-level*] [-o *optroot*] [-r]

DESCRIPTION

utparalleld provides printer support for Sun Ray appliances. utparalleld supplies driver services for all USB parallel adaptors and USB printers that comply with the USB printer class.

utparalleld uses the utparallel(7D) loopback driver to provide Solaris applications the same interface as standard workstation parallel ports, such as /dev/ecpp or /dev/bpp. Solaris applications such as the lp(1) daemon can use device nodes that are provided by utparalleld.

When a parallel adaptor or a USB printer is attached to an appliance, utparalleld creates device nodes in the /tmp/SUNWut/units/*CID* directory. *CID* is the canonical identifier for an appliance (IEEE802.*nnnnnnnnnnn).* UTDEVROOT points to this directory through a symbolic link. A user can use $UTDEVROOT/dev/printers/*printer-name* to access a specific printer.

Error messages from utparalleld are logged using syslog(3), with a facility value of LOG_DAEMON.

OPTIONS

The following options are supported.

-D *debug-level*	Debug mode. Use is beyond the scope of this document.
-o *optroot*	Use *optroot* as the parallel service's root directory for device node creation. The default value is /tmp/SUNWut. *optroot* should be the same directory as the *optroot* directory used by utdevmgrd(1M).
-r	Automatically restart the printer service daemon if it exits. With this option, the printer service daemon creates two processes: a child that performs all the actual work, and a parent monitoring process. The parent process restarts a child if the previous one exits.

FILES

The following files are used:

/tmp/SUNWut	The customary location for temporary files used by the Sun Ray server managers, designated by *optroot*.
/tmp/SUNWut/.utdevmgr	The named pipe used for communication between the device manager and device driver services.

`/tmp/SUNWut/units`	The directory containing device directories for each appliance. The directory names represent the appliances' canonical identifier. A canonical identifier has the form IEEE802.*nnnnnnnnnnn* or *nnnnnnnnnnn* (the 12-digit hexadecimal MAC address of the appliance). Each directory contains a `dev` directory and a `devices` directory.
`/tmp/SUNWut/units/`*CID*`/dev/printers`	The directory containing links to parallel device names for each appliance. *CID* is the canonical identifier for an appliance.
`/tmp/SUNWut/sessions`	The directory containing links to appliances in the `units` directory, named by X-Windows display number for each session. These links change as users move from one Sun Ray appliance to another.

ENVIRONMENT VARIABLES

UTDEVROOT points to a symbolic link of the device root for the Sun Ray appliance associated with a user's session.

ATTRIBUTES

See `attributes(5)` for descriptions of the following attributes:

Attribute Type	Attribute Value
Availability	SUNWuto

SEE ALSO

`utauthd(1M)`, `utdevmgrd(1M)`, `syslog(3)`, `syslogd(1M)`, `syslog.conf(4)`, `utparallel(7D)`

NAME

utpolicy - Sun Ray authentication manager policy management command.

SYNOPSIS

/opt/SUNWut/sbin/utpolicy -a[-g][-k *type*][-m][-M][-p][-r *type*][-s *type*][-z *type*]

/opt/SUNWut/sbin/utpolicy -a[-t {list | clear | add:*terminalID* | del:*terminalID*}]

/opt/SUNWut/sbin/utpolicy -i {-clear | -soft}

/opt/SUNWut/sbin/utpolicy -h

/opt/SUNWut/sbin/utpolicy

DESCRIPTION

The utpolicy command simplifies and writes the policy configuration of the Sun Ray authentication manager, utauthd(1M).

OPTIONS

The following options are supported.

Three categories of options are supported: Policy Setting, Card Reader Assignment, and Software Restart

-a This option, followed by valid Policy Setting, or Card Reader Assignment arguments, applies these arguments to the active authentication policy for the system. This option is not valid by itself.

POLICY SETTING

The specified Policy Setting arguments completely replace the current active authentication policy. Only arguments that are specified become active. Policy Setting and Card Reader Assignment arguments can be specified together

-g Turn on session selection within a server group. Allows the user to select on which server the user's session is run.

-k {card|pseudo|both} Enables Controlled Access Mode (CAM) for the specified session type. Selecting card enables CAM for card sessions, pseudo for terminal sessions, and both enables CAM for all types of sessions. For functionality, at least one of the -r, -s, or -z options must be invoked with the same argument as the -k option. The -k option is not considered until the utconfig application has configured for Controlled Access Mode.

`-m`	Enable multihead session capability, allowing multiple terminals to act as display devices for a single user session.
`-M`	Enable non-smart card mobile sessions.
`-p`	This option changes the behavior of the self-registration application so that it does not require the Solaris name and password before registering a token. Note that the self-registration application only verifies the name and password. They are not stored.
`-r {card|pseudo|both}`	Specify the token types that must be registered in the administrative database in order to be granted access to a login screen. Policy looks up and uses token database entry
`-s {card|pseudo|both}`	Specify the token types that will be presented with a registration screen if they do not have an entry in the administrative database. Policy allows self-registration of tokens.
`-z {card|pseudo|both}`	Specify the token types that do not require an entry in the administrative database in order to be granted access to a login screen. Policy grants access to tokens without database entry.

CARD READER ASSIGNMENT

The Card Reader Assignment arguments are incremental in nature (a complete specification does not have to occur all at once). This means a card reader can be added today and another can be added next week. Both will then be active until explicitly deleted. Policy Setting and Card Reader Assignment arguments can be specified together

`-t list`	List the terminal IDs of the Sun Ray appliances that are currently being used as dedicated card readers for registration of tokens.
`-t clear`	Reset the list of Sun Ray appliances in dedicated card reader mode.
`-t add:`*terminalID*	Add a terminal (appliance) identification to the list of terminals being used as dedicated card readers. If a partial *terminalID* is specified, then the model will be assumed to be SunRayP1. If the *terminalID* is preceded by a backslash, then the *terminalID* will be used without any transformation.
`-t del:`*terminalID*	Remove a terminal (appliance) identification from the list of terminals being used as dedicated card readers.

SOFTWARE RESTART

Software Restart options CANNOT be combined with Policy Setting or Card Reader Assignment arguments

`-i {clear | soft}` Restarts the Sun Ray services. When used with the `clear` argument, `utpolicy` clears out all existing sessions before restarting Sun Ray services. The `soft` argument leaves sessions intact. Some sessions might be unreachable after restart.

With the `-h` option, the `utpolicy` command prints out the usage message.

With no options, the `utpolicy` command prints out the policy in effect.

The following options are RESERVED for use by the Sun Ray Server Software and should not be used:

`-G,-P,-Q,-b,-f,-l,-u,-x,+x`

EXAMPLES

EXAMPLE 1 This command allows all appliances to be used with or without a smart card. Non-smart card mobile sessions are enabled.

```
# /opt/SUNWut/sbin/utpolicy -a -M -z both
```

EXAMPLE 2 This command configures the policy so that all access via smart card requires a valid administrative database entry before access is granted. If a database entry has not been created for a smart card, then a registration session is presented on the appliance. If no smart card is used, then the normal Solaris login screen is presented

```
# /opt/SUNWut/sbin/utpolicy -a -r card -s card -z pseudo
```

EXAMPLE 3 This command is like the previous except that it does not allow for users to register their own smart cards. Instead, it is assumed that the appliance specified in the `-t add:` option will be used along with the appropriate administrative tools to create the necessary database entries. In this example, the *terminalID* is expanded to `SunRayP1.080020a8e723`

```
# /opt/SUNWut/sbin/utpolicy -a -r card -z pseudo -t clear -t
add:080020a8e723
```

EXAMPLE 4 This command configures the policy so that all smart card users must be registered, however non-smart card users can still use the appliances. Smart card users can self-register. Controlled Access Mode is invoked for smart card users.

```
# /opt/SUNWut/sbin/utpolicy -k card -r card -s card -z pseudo
```

FILES

The following files are used:

- `/etc/opt/SUNWut/policy/utpolicy`
  The policy configuration file
- `/etc/opt/SUNWut/terminals`
  The list of appliances being used as dedicated card readers
- `/etc/opt/SUNWut/auth.props`
  Sun Ray authentication manager's configuration file

ATTRIBUTES

See `attributes(5)` for descriptions of the following attributes:

Attribute Type	Attribute Value
Availability	SUNWuto

SEE ALSO

`utauthd(1M)`, `auth.props(4)`

NAME `utpreserve` - Sun Ray configuration file preservation utility.

SYNOPSIS **/cdrom/cdrom0/utpreserve** [`-d` *preserve-directory*]

DESCRIPTION The `utpreserve` command stops Sun Ray services, terminates user sessions, and saves existing Sun Ray server configuration data into a compressed tar file, `/var/tmp/SUNWut.upgrade/preserve_1.3.tar.Z`.

OPTIONS The following option is supported.

`-d` *preserve-directory* Save the compressed tar file into the *preserve-directory*.

SEE ALSO `utinstall(1M)`, `utconfig(1M)`

NAME

`utpw` - Sun Ray administration password change utility.

SYNOPSIS

/opt/SUNWut/sbin/utpw

DESCRIPTION

The `utpw` command changes the Sun Ray administrator password (also known as the "UT admin" password). This password is entered by the administrator when logging into the Administration Tool and is used to make a privileged connection to the LDAP server.

`utpw` changes the password both in the administration database, and the password file on the local server. To successfully change the administration password, the system administrator must also provide the `rootdn` password. By default, the `rootdn` password is the same as the administration password.

In a failover group, `utpw` also affects the administration database of the secondary servers, but only the password file on the local server. The administrator must log into the secondary servers and run `utpw` on them to change the password files.

OPTIONS

There are no options for `utpw`.

EXAMPLES

EXAMPLE 1 This command changes the administration password:

```
# /opt/SUNWut/sbin/utpw
Enter new UT admin password:
Re-enter new UT admin password:
Enter old UT admin password:

Changing LDAP password...
Done.
Changing password file...
Done.
```

FILES

The following files are used:

- `/etc/opt/SUNWut/utadmin.pw`
- `/etc/opt/SUNWut/utadmin.conf`

EXIT STATUS

The following exit values are returned:

`0` Success

`1` Error

ATTRIBUTES

See `attributes(5)` for descriptions of the following attributes:

Attribute Type	Attribute Value
Availability	SUNWuta

SEE ALSO utdesktop(1M), utuser(1M), *Sun Ray Server Software 1.3 Administrator's Guide*

NOTES The -f option has been deprecated. Use utpw instead. If you use the -f option, you must supply the Sun Ray administration password though there is no prompt for it displayed.

NAME

`utrcmd` - Sun Ray remote administration utility.

SYNOPSIS

/opt/SUNWut/lib/utrcmd [`-n`] *hostname command* [*args*]

DESCRIPTION

The `utrcmd` program provides a way to run Sun Ray administration commands remotely. The `utrcmd` program contacts the in.utrcmdd daemon on the remote *hostname* and executes the specified *command* with the specified arguments *args* (if any).

`utrcmd` copies it's standard input to the specified command, the standard output of the command to `utrcmd`'s standard output, and the standard error of the command to `utrcmd`'s standard error. Interrupt, quit, and terminate signals are propagated to the specified command; `utrcmd` terminates normally when the command does.

OPTIONS

The following option is supported.

`-n` Redirect the input of `utrcmd` to `/dev/null`. This option prevents interactions between `utrcmd` and the shell which invokes it. For example, if you are running `utrcmd` and invoke a `utrcmd` in the background without redirecting its input away from the terminal, it will block even if no reads are posted by the specified command. The `-n` option prevents this behavior.

USAGE

Official hostnames or nicknames may be given as the *hostname*.

The `utrcmd` and `in.utrcmdd` programs use the Sun Ray failover group configuration to perform a set of checks before allowing the specified command to proceed.

The program `utrcmd` runs with set-user-ID permission of root or superuser. The `utrcmd` command will only proceed if all of the following are true (on the initiating system):

- The user's real user-ID is superuser, or the user has membership rights in the `utadmin` group.
- The `auth.props` file is owned by superuser and is not writable by anyone other than superuser.
- The `gmSignatureFile` property of `auth.props` specifies a group signature file.
- The group signature file exists and is owned by superuser and is not readable, writable, or executable by anyone other than superuser.
- The group signature file is at least 8 bytes long and has similar content diversity characteristics as required by `passwd`(1).
- The "`utrcmd/tcp`" service is enabled.

The in.utrcmdd program will accept the connection only if all of the following are true (on the remote system):

- The utrcmd/tcp service is enabled and matches the configuration on the initiating system.
- The in.utrcmdd program is enabled in /etc/inetd.conf.
- The utadmin group is configured on the system.
- The auth.props file is owned by superuser and is not writable by anyone other than superuser.
- The gmSignatureFile property of auth.props specifies a group signature file.
- The group signature file exists and is owned by superuser and is not readable, writable, or executable by anyone other than superuser.
- The group signature file is at least 8 bytes long and has similar content diversity characteristics as required by passwd(1).

If the connection is accepted, the utrcmd program begins a challenge-response handshake with the in.utrcmdd program, using the contents of the group signature file to sign messages (without revealing the contents of the signature file). Either utrcmd or in.utrcmdd will reject the transaction if the handshake fails. The specified command will not be run if the contents of the group signature files on the two systems differ.

Finally, in.utrcmdd will reject the specified command if it is not recognized or if the command or it's arguments contain disallowed characters (such as '[;']'), which may cause a security problem while interpreting the command. Specified commands always run in group utadmin.

The following are specified commands:

- /opt/SUNWut/sbin/utpolicy
- /opt/SUNWut/sbin/utfwadm
- /usr/sbin/dhtadm
- /usr/sbin/pntadm

EXAMPLES

EXAMPLE 1 This command lists the configured token readers on a remote Sun Ray server

```
# /opt/SUNWut/lib/utrcmd sun5 /opt/SUNWut/sbin/utpolicy -t list
```

FILES

The following files are used:

- /etc/hosts

  Internet host table
- /etc/group

Group file

- `/etc/inet/services`

  Internet services table

- `/etc/inetd.conf`

  Internet services daemon configuration table

- `/etc/opt/SUNWut/auth.props`

  Sun Ray authentication properties file

ATTRIBUTES

See `attributes(5)` for descriptions of the following attributes:

Attribute Type	Attribute Value
Availability	SUNWuto

SEE ALSO

`utauthd(1M)`, `inetd(1M)`, `group(4)`, `auth.props(4)`, `hosts(4)`, `nsswitch(4)`, `passwd(1)`, `rsh(1)`, `attributes(5)`

NOTES

`utrcmd` works in a manner similar to `rsh(1)`. However, it imposes multiple restrictions to maintain system security.

NAME

`utreplica` - LDAP replication utility for Sun Ray servers.

SYNOPSIS

/opt/SUNWut/sbin/utreplica `-p` *secondary-server1* [*secondary-server2*]...

/opt/SUNWut/sbin/utreplica `-s` *primary-server*

/opt/SUNWut/sbin/utreplica `-l`

/opt/SUNWut/sbin/utreplica `-u`

DESCRIPTION

The `utreplica` command configures the Sun Ray LDAP server to replicate data from the primary server to each secondary server in a failover group. The command must be run with superuser privileges on the Sun Ray server to be configured.

OPTIONS

The following options are supported.

`-l`	List the current failover administration status.
`-p` *secondary-server*	Configure the primary server. *secondary-server* is the host name of the secondary server. List all seondary servers within the failover group.
`-s` *primary-server*	Configure a secondary server. *primary-server* is the host name of the primary server.
`-u`	Unconfigure this Sun Ray server for LDAP database replication.

USAGE

`utreplica` is used only on Sun Ray servers in a failover group. Configure the primary Sun Ray server first, then the secondary servers.

FILES

The following files are configured on the primary Sun Ray server:

- `/etc/opt/SUNWconn/ldap/current/dsserv.conf`
- `/etc/opt/SUNWconn/ldap/current/dsserv.ini`

The following files are configured on the secondary Sun Ray server:

- `/etc/opt/SUNWconn/ldap/current/dsserv.conf`
- `/etc/opt/SUNWut/utadmin.conf`

ATTRIBUTES

See `attributes(5)` for descriptions of the following attributes:

Attribute Type	Attribute Value
Availability	SUNWuto

SEE ALSO

`utconfig(1M)`

NOTES

To replicate the LDAP information properly, all Sun Ray servers in the failover group must have the same group signature.

Use of the `-p`, `-s`, and `-u` options on a Sun Ray server will interrupt all active sessions on that server.

NAME

`utselect` - Sun Ray failover group server selection tool.

SYNOPSIS

/opt/SUNWut/bin/utselect [`-L`] [`-R`] [`-S`] [`-X`]

DESCRIPTION

The `utselect` command is a graphical user interface (GUI) to the `utswitch` command. It allows user selection of a Sun Ray server or session for the Sun Ray appliance to connect. The sessions in the GUI are sorted in order of most current. The second item in the list is highlighted by default to allow easy switching between two servers. The Refresh button executes the `utswitch -l` command and updates the information displayed in the GUI. The Ok button executes a `utswitch -h` command to the server highlighted.

OPTIONS

The following options are supported.

`-L` Configures `utselect` to run in "login" mode before the CDE log in screen is displayed. Where:

- If only one server is available, the command exits.
- The current server is set as the default.
- Selecting the current server causes the command to exit
- The locale is determined in a manner similar to CDE
- The screen is centered in the display

`-R` Remote server selection is enabled. This enables an entry field where a networked server name can be entered.

`-S` Remote server selection is disabled.

`-X` Exit after making a selection from the list.

EXAMPLES

EXAMPLE 1 This command enables users to select which Sun Ray server or session to connect. The GUI will exit after selection

```
% /opt/SUNWut/bin/utselect -X
```

ATTRIBUTES

See `attributes(5)` for descriptions of the following attributes:

Attribute Type	Attribute Value
Availability	SUNWuto

SEE ALSO

`utswitch(1)`, `attributes(5)`

NAME

`utserial, utparallel` - Sun Ray serial and parallel port device driver emulators.

SYNOPSIS

```
#include <sys/types.h>
#include <fcntl.h>
```

utserial

```
#include <sys/termios.h>
#include <termio.h>
```

utparallel

```
#include <sys/ecppio.h>
```

DESCRIPTION

`utserial` is a tty-style interface that provides a generic virtual interface to USB serial adaptors connected to the Sun Ray appliance.

`utparallel` is a parallel-style interface that provides a generic virtual interface to USB parallel adaptors connected to the Sun Ray appliance.

`utserial` and `utparallel` are each loadable `STREAMS` drivers.

EXTENDED DESCRIPTION

The actual interface to the appliance for each of these drivers is through the Sun Ray interconnect via either the `utseriald` daemon or the `utparalleld` daemon, each of which is session-aware. The daemons are connected to either `utserial` or `utparallel` through a master port and each is responsible for creating the slave device nodes through which normal applications will connect.

API

Applications open a device file created by either `utseriald` or `utparalleld`. Device files created by `utseriald` comply to the `termio`(7I) interface and device files created by `utparalleld` comply to the `ecpp`(7D) interface. Hardware limitations in USB adaptors might prevent compliance with these interfaces.

FILES

The following files are used:

- `/dev/utserial`

  Master port for `utserial`
- `/dev/utparallel`

  Master port for `utparallel`

ATTRIBUTES

See `attributes(5)` for descriptions of the following attributes:

Attribute Type	Attribute Value
Availability	SUNWutu
MT-Level	Safe

SEE ALSO utseriald(1M), utparalleld(1M), termio(7I), ecpp(7D)

NAME

`utseriald` - Sun Ray appliance serial services daemon.

SYNOPSIS

/opt/SUNWut/bin/utseriald [`-D` *debug-level*] [`-o` *optroot*] [`-r`]

DESCRIPTION

`utseriald` provides serial support for Sun Ray appliances through driver services for USB serial adaptors. For a list of supported adaptors, see the web site:

`http://www.sun.com/sunray`

`utseriald` uses the `utserial`(7D) loopback driver to provide Solaris applications the same interface as standard workstation serial ports, such as `/dev/term/a` or `/dev/term/b`. Solaris applications such as `tip(1)` can change port settings through the standard `termio`(7I) interface.

When an application opens a device node, the `ldterm`(7M) and `ttcompat(7M)` modules are pushed below the STREAM head.

When a serial adaptor is attached to an appliance, `utseriald` creates device nodes in the `/tmp/SUNWut/units/`*CID* directory. *CID* is the canonical identifier for an appliance (IEEE802.*nnnnnnnnnnn).* `UTDEVROOT` points to this directory through a symbolic link. A user can use `$UTDEVROOT/dev/term/`*terminal-name* to access the serial ports on the Sun Ray appliance.

Error messages from `utseriald` are logged using `syslog`(3), with a facility value of `LOG_DAEMON`.

OPTIONS

The following options are supported.

`-D` *debug-level* Debug mode. Use is beyond the scope of this document.

`-o` *optroot* Use *optroot* as the serial service's root directory for device node creation. The default value is `/tmp/SUNWut`. *optroot* should be the same directory as the *optroot* directory used by `utdevmgrd`(1M).

`-r` Automatically restart the serial service daemon if it exits. With this option, the serial service daemon creates two processes: a child that performs all the actual work, and a parent monitoring process. The parent process restarts a child if the previous one exits.

FILES

The following files are used:

`/tmp/SUNWut` The customary location for temporary files used by the Sun Ray server managers, designated by *optroot*.

`/tmp/SUNWut/.utdevmgr` The named pipe used for communication between the device manager and device driver services.

/tmp/SUNWut/units	The directory containing device directories for each appliance. The directory names represent the appliances' canonical identifier. A canonical identifier has the form IEEE802.*nnnnnnnnnnn* or *nnnnnnnnnnn* (the 12-digit hexadecimal MAC address of the appliance). Each directory contains a dev directory and a devices directory.
/tmp/SUNWut/units/*CID*/dev/term	The directory containing links to serial device names for each appliance. *CID* is the canonical identifier for an appliance.
/tmp/SUNWut/sessions	The directory containing links to appliances in the units directory, named by X-Windows display number for each session. These links change as users move from one Sun Ray appliance to another.

ENVIRONMENT VARIABLES

UTDEVROOT points to a symbolic link of the device root for the Sun Ray appliance associated with a user's session.

ATTRIBUTES

See attributes(5) for descriptions of the following attributes:

Attribute Type	Attribute Value
Availability	SUNWuto

SEE ALSO

tip(1), utauthd(1M), utdevmgrd(1M), utserial(7D),ldterm(7M), ttcompat(7M), termio(7I), zs(7D), sad(7D), syslog(3), syslogd(1M), syslog.conf(4)

NAME

`utsession` - List and manage the Sun Ray sessions on the local Sun Ray server.

SYNOPSIS

/opt/SUNWut/sbin/utsession `-p` [`-x`] [`-d` *disp#*] [`-u` *unix*] [`-t` *token*] [`-n` *name*]

/opt/SUNWut/sbin/utsession {`-k` | `-s`} [`-a`] [`-x`] { [`-d` *disp#*] [`-u` *unix*] [`-t` *token*] [`-n` *name*]}

/opt/SUNWut/sbin/utsession `-r` [`-a`] [`-x`] [`-d` *disp#*] [`-u` *unix*] [`-t` *token*] [`-n` *name*]

/opt/SUNWut/sbin/utsession `-h`

DESCRIPTION

The first synopsis (`-p`) is used to print Sun Ray sessions (both active and suspended) for the specified user or token on the current server. When the `-u`, `-t`, `-n`, and `-d` options are not used, `utsession` prints all of the Sun Ray sessions on the current server.

When listing the Sun Ray sessions, `utsession` also lists the state for each session:

D	Disconnected — The session is currently not attached to any Sun Ray. The session is considered connected if this flag is omitted.
I	Idling — A `dtlogin` session that is currently waiting for user login (ie. `dtgreet`). User has already logged into the `dtlogin` session if the flag is omitted.
S	Suspended — The session is currently suspended. The session is considered running if this flag is omitted.

The second synopsis (`-k | -s`) is used to kill or suspend Sun Ray sessions on the current server. At least one of the `-d`, `-u`, `-t`, or `-n` options must be specified. Unless the `-a` option is specified, more than one session matching the specified criteria will return an error.

When suspending a session, `utsession` uses `dterror.ds(1)` to display a message on the user's screen to indicate that the session is being suspending by the administrator.

The third synopsis (`-r`) is used to resume Sun Ray sessions on the current server. If `-d`, `-u`, `-t`, or `-n` option is not specified, `utsession` will resume all the suspended Sun Ray sessions on the current server. Unless the `-a` option is specified, more than one session matching the specified criteria will return an error.

The fourth synopsis (`-h`) displays the usage of this command.

OPTIONS

The following options are supported:

`-a`	Apply the operation to all matching sessions if more than one matches the search criteria. If `-a` is not specified, multiple matching sessions return an error.
`-d` *disp#*	Specify the X display number for search.
`-h`	Display the usage of this command.
`-k`	Kill the sessions matching the search criteria. You must also specify at least one of the `-d`, `-u`, `-t`, or `-n` options.
`-n` *name*	Specify the registered Sun Ray username for search. Sessions belonging to users matching the username are listed. It is a case sensitive, exact match.
`-p`	Print the sessions belonging to the specified user or token. If any matching criteria is specified, it will include the matched suspended sessions.
`-r`	Make the sessions matching the search criteria active. `utsession` will resume all specified Sun Ray sessions if `-d`, `-u`, `-t`, or `-n` options are not specified.
`-s`	Suspend the sessions matching the search criteria. You must also specify at least one of the `-d`, `-u`, `-t`, or `-n` options.
`-t` *token*	Specify the Sun Ray token for search. The token is in one of the following forms: ■ Raw token form for unregistered users (`MicroPayflex.`*####*) ■ Pseudo token form for terminal users (`pseudo.`*macaddr*) ■ Logical token form for registered users (`user.`*#####*) ■ Mobile token form for NSC mobile users (`mobile.`*username*)
`-u` *unix*	Specify the UNIX login name for search.
`-x`	RESERVED for special handling. This is invoked by `utrcmd(1M)` to support remote operation of the Administration Tool interface.

EXAMPLES

EXAMPLE 1 This command lists all sessions on the current server.

```
# utsession -p
```

EXAMPLE 2 This command finds the sessions for the UNIX user "jdoe".

```
# utsession -p -u jdoe
```

EXAMPLE 3 This command terminates a registered Sun Ray user's (john doe's) session.

```
# utsession -k -n "john doe"
```

EXAMPLE 4 This command suspends the session belonging to pseudo token `MicroPayflex.000105d665000100` on display `10`.

```
# utsession -s -d 10 -t MicroPayflex.000105d665000100
```

FILES

The following files are used:

- `/etc/opt/SUNWut/utsession.msg`

  The message displayed when a session is suspended. Text editable and localizeable.

EXIT STATUS

The following exit values are returned:

`0`	Command completed successfully.
`1`	Entry not found.
`-1`	An error occurred.

ATTRIBUTES

See `attributes(5)` for descriptions of the following attributes:

Attribute Type	Attribute Value
Availability	SUNWuta

SEE ALSO

`utuser(1M)`, `auth.props(4)`, `utdesktop(1M)`

NAME

`utsessiond` - Sun Ray session manager daemon.

SYNOPSIS

/opt/SUNWut/lib/utsessiond [`-a` *authlist*] [`-c` *authfile*] [`-d`] [`-h` *hostname*] [`-p` *port*] [`-P` *nport*] [`-r`] [`-t`]

DESCRIPTION

The `utsessiond` daemon provides a reliable rendezvous point for services in a Sun Ray session. It acts as an intermediary to forward session connection and disconnection messages from the Sun Ray authentication manager to the services and provides facilities for supporting distributed synchronization of clip-lists for the services.

If either the `-a` or the `-c` options is specified, the session manager daemon operates exclusively in call-back mode. In this mode, the session manager only takes session connect and disconnect commands from authentication managers that are explicitly enabled by *authlist* or *authfile* and that have requested a call-back. The call-back feature provides a mechanism by which the session manager and the authentication manager may establish each other's identity.

Error messages from `utsessiond` are logged using `syslog`(3) with a facility value of LOG_DAEMON.

OPTIONS

The following options are supported.

`-a` *authlist* — Add the host and port pairs specified in *authlist* to the list of permitted Authentication Managers. The format of *authlist* is a comma separated list of *hostname*:*port* pairs.

`-c` *authfile* — Add the host and port pairs specified in the ASCII file *authfile* to the list of permitted Authentication Managers. The file contains a list of Authentication Manager specifications, one per line. The specifications take the form of *hostname* followed by *port* number, separated by white-space. Blank lines and any line whose first printable character is "#" are ignored.

`-d` — Enable debugging output.

`-h` *hostname* — Set the *hostname* portion of the session IDs generated by the Session Manager to the specified *hostname* value. By default this is set to the machine's node name. This option can be used to handle servers supporting multiple IP addresses as part of a clustering solution.

`-p` *port* — Set the Session Manager's listen port to the specified *port* value. The default is port 7007. This is the port by which session services and Authentication Managers contact the Session Manager.

`-P` *nport* — This option is no longer used. Retained only for backward compatibility.

-r	Automatically restart the Session Manager daemon if it exits. With this option the Session Manager daemon will create two processes: a child that performs all the actual work and parent monitoring process. The parent process will restart a child if the previous one exits.
-t	Test mode. Use is beyond the scope of this document.

FILES

The following file is used:

- /etc/opt/SUNWut/auth.permit

  The customary location of the authfile for a system.

ATTRIBUTES

See attributes(5) for descriptions of the following attributes:

Attribute Type	Attribute Value
Availability	SUNWuto

SEE ALSO

utauthd(1M), syslog(3), syslogd(1M), syslog.conf(4)

NAME

`utsettings` - View or change the Sun Ray appliance settings.

SYNOPSIS

/opt/SUNWut/bin/utsettings [`-H` [`-k` *hotkey*]]

DESCRIPTION

The `utsettings` command opens a Sun Ray Settings dialog box that allows the user to view or change audio, visual, and tactile interface settings for the Sun Ray appliance.

The `utsettings` application connects to the Session Manager and is informed of which appliance it is being displayed upon. As the user moves a session from one Sun Ray appliance to another, the Session Manager keeps track of the session's current location and instructs the `utsettings` application to follow. With each session move, the `utsettings` application displays the current appliance's configuration.

By default, the Sun Ray server starts an instance of `utslaunch(1M)` for each session created by the user logging in via `dtlogin`. This enables the Sun Ray Settings dialog box to be available by pressing a hotkey or key combination. Subsequent presses toggle the dialog box on and off.

Users can initiate similar functionality by running `utsettings` with the `-H` flag. The hotkey can be specified using the `-k` option. Only one instance of `utsettings -H` or `utslaunch` can be running per session.

OPTIONS

The following options are supported.

`-H` Start the `utsettings` application in "hotkey" mode. The `utsettings` application starts with the Sun Ray Settings dialog box hidden. Pressing the hotkey toggles the display of the dialog box. The default hotkey key combination is Shift + Props. The hotkey can be user or site defined according to the `utsettings.hotkey` property in the files listed below. See FILES

`-k` *hotkey* Use the specified key or keys as the hotkey combinations when the `-H` option is specified. This option is dependent upon the `-H` option.

EXAMPLES

EXAMPLE 1 This command displays the settings for the Sun Ray appliance you are currently logged into.

```
% utsettings
```

EXAMPLE 2 This command displays the settings for the Sun Ray appliance you are currently logged into when you press the Shift + ESC hotkey key combination.

```
% utsettings -H -k "Shift ESC"
```

FILES

The following files are used:

- /etc/opt/SUNWut/utsettings_defaults.properties
  site-wide defaults
- ~/.utsettings.properties
  user's defaults
- /etc/opt/SUNWut/utsettings_mandatory.properties
  site-wide mandatory defaults

EXIT STATUS

The following exit values are returned:

0	Success
1	Failure

ATTRIBUTES

See attributes(5) for descriptions of the following attributes:

Attribute Type	Attribute Value
Availability	SUNWuto

SEE ALSO

utslaunch(1M), dtlogin(1X), dtsession(1X), utsettings.properties(4)

NAME

`utsettings.properties` - Default values for the `utsettings` application.

SYNOPSIS

/etc/opt/SUNWut/utsettings_defaults.properties

~/.utsettings.properties

/etc/opt/SUNWut/utsettings_mandatory.properties

DESCRIPTION

The files listed above are standard Java properties files that can contain defaults which customize the operation of the `utsettings` application. Each file contains entries in the format of:

name=value

where *name* is the property name and *value* is the value set.

None of the properties files are required for correct operation of the `utsettings` application. If none are present, the application will use internal defaults.

EXTENDED DESCRIPTION

When the application starts, it looks for and reads the properties files in the order listed below. A property specified within a file can be overridden by a file read later in the search order.

1. `/etc/opt/SUNWut/utsettings_defaults.properties`

   This file contains site-wide default properties which are used if the user has not specified any. These properties override any application defaults.

2. `~/.utsettings.properties`

   This file contains the user's default properties. These properties override application and site-wide default properties.

3. `/etc/opt/SUNWut/utsettings_mandatory.properties`

   This file contains site-wide mandatory default properties which supersede any application, site-wide, or user defaults.

PROPERTIES

The supported application properties are listed below. For each property, the name, description, application default, and some examples are given.

Name — `utsettings.hotkey`

Description — Specifies the hotkey or key combination that invokes the Sun Ray Settings dialog box. The value is a valid X keysym name preceded by one or more of the supported modifiers (Ctrl, Shift, Alt, Meta).

Application Default — Shift SunProps (Hold down Shift and press the Props key)

Examples:

- F3

- Shift F4
- Ctrl Shift Alt F5

EXAMPLES

EXAMPLE 1 The following is the sample contents of a properties file. The values shown below are the application defaults that would be in effect if no properties files existed.

```
utsettings.hotkey=Shift SunProps
```

ATTRIBUTES

See `attributes(5)` for descriptions of the following attributes:

Attribute Type	Attribute Value
Availability	SUNWutr

SEE ALSO

`utsettings(1)`, `utslaunch(1M)`, `utslaunch.properties(4)`

NOTES

F11 and F12 are not valid values for `utsettings.hotkey`.

Consider that the hotkey property definition in the properties files are deprecated as of the SRSS 1.3 release. In the future, this will be defined in the `utslaunch.properties` file. Please refer to its man page and discontinue this usage as it will not be supported in the next Sun Ray server software release.

NAME

`utslaunch` - Sun Ray appliance launch application.

SYNOPSIS

/opt/SUNWut/lib/utslaunch

DESCRIPTION

The `utslaunch` application is used to launch various Sun Ray applications via a "hotkey" key combination. The applications are enabled when the key combination is pressed.

The `utslaunch` application provides hotkey functionality while consuming fewer system resources.

Hotkey key combinations are defined in the `utslaunch.properties` files.

OPTIONS

There are no options for `utslaunch`.

EXAMPLES

EXAMPLE 1 This command starts the `utslaunch` daemon in the background.

```
# utslaunch &
```

FILES

The following files are used for hotkey configuration:

- `/etc/opt/SUNWut/utslaunch_defaults.properties`
  site-wide defaults
- `~/.utslaunch.properties`
  user's defaults
- `/etc/opt/SUNWut/utslaunch_mandatory.properties`
  site-wide mandatory defaults

The following file is used:

`/usr/dt/config/Xsession.d/0100.SUNWut`

ENVIRONMENT VARIABLES

`utslaunch` uses the `DISPLAY` environment variable to get the user's X display number.

It also uses the `HOME` environment variable to get the user's home directory to be able to use user's hotkey settings.

EXIT STATUS

The following exit values are returned:

`0`	Success
`1`	Failure

ATTRIBUTES

See attributes(5) for descriptions of the following attributes:

Attribute Type	Attribute Value
Availability	SUNWuto

SEE ALSO

utslaunch.properties(4), utsettings(1), utsettings.properties(4), utdetach(1)

NAME

`utslaunch.properties` - Default hotkey key combinations for various applications supported by `utslaunch`.

SYNOPSIS

/etc/opt/SUNWut/utslaunch_defaults.properties

~/.utslaunch.properties

/etc/opt/SUNWut/utslaunch_mandatory.properties

DESCRIPTION

The files listed above are standard Java properties files that can contain defaults which customize the operation of the `utslaunch` application. Each file contains entries in the format of:

name=value

where *name* is the property name and *value* is the value set.

EXTENDED DESCRIPTION

When the `utslaunch` application starts, it looks for and reads the properties files in the order listed below. Note that a hotkey key combination specified in a file can be overridden by a file read later in the search order.

1. `/etc/opt/SUNWut/utslaunch_defaults.properties`

   This file contains site-wide default properties which are used if the user has not specified any. These properties override any application defaults.

2. `~/.utslaunch.properties`

   This file contains the user's default properties. These properties override application and site-wide default properties.

3. `/etc/opt/SUNWut/utslaunch_mandatory.properties`

   This file contains site-wide mandatory default properties which supersede any application, site-wide, or user defaults.

PROPERTIES

The supported application properties are listed below. For each property, the name, description, application default, and some examples are given.

Name — `utdetach.hotkey`

Description — Specifies the hotkey or key combination that disconnects the current session from the appliance the user is currently using. The value is a valid X keysym name preceded by one or more of the supported modifiers (Ctrl, Shift, Alt, Meta).

Application Default — Shift Pause (Hold down Shift and press the Pause key)

Name — `utsettings.hotkey`

Description — Specifies the hotkey or key combination that invokes the Sun Ray Settings dialog box. The value is a valid X keysym name preceded by one or more of the supported modifiers (Ctrl, Shift, Alt, Meta).

Application Default — Shift SunProps (Hold down Shift and press the Props key)

Examples:

- F3
- Shift F4
- Ctrl Shift Alt F5

EXAMPLES

EXAMPLE 1 The following is a sample of the contents of a properties file.

```
utdetach.hotkey=Shift Pause
utsettings.hotkey=Shift SunProps
```

ATTRIBUTES

See `attributes(5)` for descriptions of the following attributes:

Attribute Types	Attribute Values
Availability	SUNWutr
Stability Level	Evolving

SEE ALSO

`utslaunch(1M)`, `utdetach(1M)`, `utsettings(1M)`

NOTES

The property file entry for `utsettings.hotkey` deprecates the `utsettings.property` file entry for the same property. The `utsettings.property` file entry will be removed in the next release. Please use this new `utslaunch.properties` interface from Sun Ray server software 1.3 onwards.

NAME

`utsunmc` - Sun Ray server software 1.3 module for the Sun Management Center, providing addition, load, and removal utilities.

SYNOPSIS

/opt/SUNWut/sbin/utsunmc [`-u`]

DESCRIPTION

The `utsunmc` command adds the Sun Ray server software 1.3 module to the Sun Management Center (SunMC) and loads it to permit monitoring of the Sun Ray software. The `utsunmc` command can also remove the Sun Ray server software 1.3 module from the SunMC.

The `utsunmc` command is run with superuser privileges.

OPTIONS

The following option is supported.

`-u` Remove the previously added and loaded Sun Ray server software 1.3 module.

Without arguments, addition and load of the Sun Ray server software 1.3 module is performed.

ATTRIBUTES

See `attributes(5)` for descriptions of the following attributes:

Attribute Type	Attribute Value
Availability	SUNWutesa

SEE ALSO

Sun Management Center 3.0 Software User's Guide

NOTES

The `utsunmc` command requires the Sun Management Center 3.0 or Sun Management Center 2.1.1 to be installed.

The Sun Management Center agent is stopped while this command runs and restarted after the command completes. The agent may not properly restart on Sun Management Center 2.1.1. In this case, the command `/opt/SUNWsymon/sbin/es-start -a` should be run.

NAME

`utswitch` - Sun Ray server selection and session listing utility.

SYNOPSIS

/opt/SUNWut/bin/utswitch {`-l` | `-t` | `-h` *hostname*} [`-k` *token*] [`-p` *port*] [`-r`] [`-s` *sid*]

DESCRIPTION

The `utswitch` command allows switching a Sun Ray appliance among Sun Ray servers in a failover group. It can also list the existing sessions for the current token. One of the following option flags must be specified: `-l`, `-t`, or `-h`. The `utselect`(1) command implements a GUI-based interface to this command.

OPTIONS

The following options are supported.

`-h` *hostname* Force an explicit switch to the server with *hostname*.

`-k` *token* Specify the token ID *token* to be used in collecting session information from the servers in the failover group. The token normally used is the one connected to the current session.

`-l` List the servers accessible from the current Sun Ray appliance for the current token and show any existing sessions on those servers.

- The first field of the output is the server name.
- The second field is the X display number for an active user session. If no active user session exists, then -1 is printed or -2 is printed if the login screen is being displayed.
- The third field is the last connection time to an existing session, as a time value from the `time(2)` system call. If there is no session, the third field indicates status from the host as:
  - -1 Server is up, but there is no session.
  - -2 No response received from the server.
  - -3 No path from the Sun Ray to the server.
- The fourth field is 1 if the server is offline and 0 otherwise.

`-p` *port* Sets the port number of the Authentication Manager on the Sun Ray server to *port*, instead of the default 7009.

`-r` Forces a remote redirection outside of the current failover group to search for an existing session within an external failover group. If no session is available, load balancing is performed. Without this option, the Sun Ray appliance is bound explicitly to the target Sun Ray server, rather than to an appropriate server within the target failover group. This option may only be used with the `-h` option

`-s` *sid*
: Specifies the session ID *sid* of a session connected to a Sun Ray appliance and perform the requested operation on that appliance. The default is to use the session ID of the current session. Since session IDs other than that of the current session are only available to `root`, this option is not useful for a general user.

`-t`
: Switch to the server whose session has the latest connection time among the existing sessions for the current token. Normally this would switch to the current session, so it has limited usefulness. However, it is useful in the case of logging out of an existing X session and back to the login screen. The connection time of a logged out session is biased back in time so that the session will not be selected if there is an existing logged-in X session on another server. From a CDE login screen, it is possible to force a call to `utswitch -t` by selecting Reset Login Screen from the Options menu. This allows switching back to a logged-in session from the login screen without having to log in.

FILES

The following files are used:

- `/var/opt/SUNWut/displays/*`
  X display files

ATTRIBUTES

See `attributes(5)` for descriptions of the following attributes:

Attribute Type	Attribute Value
Availability	SUNWuto

SEE ALSO

`utselect(1)`, `attributes(5)`

NAME

`utuser` - Sun Ray user administration utility.

SYNOPSIS

/opt/SUNWut/sbin/utuser `-a` *"tokenID,server-name,server-port,name,other-info"* [`-r` *tokenreader*]

/opt/SUNWut/sbin/utuser `-a -f` *filename* [`-r` *token-reader*]

/opt/SUNWut/sbin/utuser `-a -i` *current-tokenID new-tokenID* [`-r` *token-reader*]

/opt/SUNWut/sbin/utuser `-d` *tokenID*

/opt/SUNWut/sbin/utuser `-d -f` *filename*

/opt/SUNWut/sbin/utuser `-d -i` *current-tokenID*

/opt/SUNWut/sbin/utuser `-e` *"tokenID,server-name,server-port,name,other-info"*

/opt/SUNWut/sbin/utuser `-e -f` *filename*

/opt/SUNWut/sbin/utuser `-e -i` *current-tokenID* [`enable | disable`]

/opt/SUNWut/sbin/utuser `-h`

/opt/SUNWut/sbin/utuser `-l`

/opt/SUNWut/sbin/utuser `-l -c`

/opt/SUNWut/sbin/utuser `-l -i` *substring*

/opt/SUNWut/sbin/utuser `-l -n` *substring*

/opt/SUNWut/sbin/utuser `-L`

/opt/SUNWut/sbin/utuser `-L -c`

/opt/SUNWut/sbin/utuser `-L -i` *substring*

/opt/SUNWut/sbin/utuser `-L -n` *substring*

/opt/SUNWut/sbin/utuser `-o`

/opt/SUNWut/sbin/utuser `-p` *tokenID*

/opt/SUNWut/sbin/utuser `-r` *token-reader*

DESCRIPTION

The `utuser` command allows the administrator to manage users registered on the Sun Ray server which the command is run. The information that `utuser` provides is from the Sun Ray administration database and the Sun Ray Authentication Manager.

`utuser` operations that only display information may be run by any user. Operations that change or delete data are run under superuser privileges.

OPTIONS

The following options are supported.

`-a`	Add user with the specified *tokenID*, *servername*, *serverport*, *name* and *otherinformation* properties. The 5 comma-delimited values should be enclosed within quotes. The *other-information* property is optional.
`-a -f`	Batch add multiple users using input from the specified *filename*. The format of each line in the input file is: *tokenID,server-name,serverport,name,other-info*
`-a -i`	Add the specified *new-tokenID* to the user that currently has token *current-tokenID*.
`-d`	Delete the user with the specified *tokenID*. This command deletes the user and all of the user's tokens. (To delete a single token without deleting the user, use the `-di` option.)
`-d -f`	Batch delete multiple users using input from the specified *filename*. The format of each line in the input file is: *tokenID*. However, you may use the output of the `-o` option as input to this option as all arguments after the first comma are ignored. For each token-id specified in the filename, this command deletes the associated user and all of the user's tokens. (To delete a single token without deleting the user, use the `-di` option.)
`-d -i`	Delete token *current-tokenID* from the user that currently has ownership of it. The token to be deleted must not be the user's only token. This command does not delete the user or any of the user's other tokens. (To delete the user and all the user's tokens, use the `-d` option.)
`-e`	Edit properties for the user with the specified *tokenID* by changing the *server-name*, *server-port*, *name* and *other-information* properties to the specified values. Note that the 5 comma-delimited values should be enclosed within quotes. The other information property is optional.
`-e -f`	Batch edit multiple users using input from the specified *filename*. The format of each line in the input file is: *tokenID,server-name,server-port,name,other-info*.
`-e -i`	Enable or disable the specified *current-tokenID*.
`-h`	Show usage information (this message).
`-l`	List all users.
`-l -c`	List all users that are currently logged in.
`-l -i`	List all users with token-ids that contain the specified substring.

`-l -n`	List all users with names that contain the specified substring.
`-L`	List all users (long format).
`-L -c`	List all users that are currently logged in (long format).
`-L -i`	List all users with token-ids that contain the specified *substring* (long format).
`-L -n`	List all users with names that contain the specified *substring* (long format).
`-o`	Dump user list in comma-delimited format. The format of each line output by this option is: *tokenID*, *server-name*, *server-port*, *name*, and *other-info*.
`-p`	Show user properties for user with the specified *tokenID*.
`-r`	When specified alone, this option reads a token-id from the specified token reader. When specified with the `-a`, `-af` or `-ai` options, the `-r` flag instructs `utuser` to use the specified token reader to assist in adding users or tokens whenever the character "x" is used in place of a token-id. The command will prompt you to insert the token into the specified reader when its ready to read the token.

For the `-l -i`, `-l -n`, `-L -i`, and `-L -n` options, the substring comparisons are case-insensitive.

EXAMPLES

EXAMPLE 1 This command displays all users that have “parker” in their usernames:

```
% /opt/SUNWut/sbin/utuser -l -n parker
```

EXAMPLE 2 This command adds a user with unknown token-ID, server name "localhost", server port "7007", user name "John Anderson", and other information "C987" by using the token reader 08002086e18f to read the token-ID:

```
# /opt/SUNWut/sbin/utuser -a "x,localhost,7007,John
Anderson,C987" -r 08002086e18f
```

EXAMPLE 3 This command adds multiple users using input from the /tmp/users file:

```
# /opt/SUNWut/sbin/utuser -a -f /temp/users
```

EXAMPLE 4 This command reads a token from token reader 08002086e18f:

```
# /opt/SUNWut/sbin/utuser -r 08002086e18f
```

FILES

The following file is used:

- /etc/opt/SUNWut/utadmin.conf

ATTRIBUTES

See attributes(5) for descriptions of the following attributes:

Attribute Type	Attribute Value
Availability	SUNWuta

SEE ALSO

utdesktop(1M), utadmin.conf(4)

NOTES

The -G option has been deprecated in favor of using utuser -L -g.

The -k option has been deprecated. Use utsession -k instead.

NAME

`utwall` - Sun Ray user notification utility.

SYNOPSIS

/opt/SUNWut/sbin/utwall `-a` *aufile* [`-r` *n*] [`-v`]{`ALL` | user [`:`*display*] | `:`*display*...}

/opt/SUNWut/sbin/utwall [`-d`][`-m` "*subject*"][`-t` "*message-text*"][`-v`]{`ALL` | user [`:`*display*] | `:`*display*...}

/opt/SUNWut/sbin/utwall [`-u` "*message-text*"]

DESCRIPTION

`utwall` sends a message or an audio file to users having an `Xsun` process. The messages can be sent in email and/or displayed in a pop-up window. When sent to a multihead session, the pop-up window will appear on all displays for that session.

Options `-a` and `-d` require superuser privileges.

OPTIONS

The following options are supported.

`-a` *aufile*	Annunciate mode. Plays the audio file *aufile* on the specified user's X session. Audio files of type `.au` can be found at `/usr/demo/SOUND/sounds`.
`-d`	Pop up a `dterror.ds` window with the supplied message on each Xsun instance.
`-m` "*subject*"	Send mail with the given subject "*subject*" and supplied message. If the text has white space, use single or double quotes. Substitution is supported.
`-r` *n*	Repeat the annunciation *n* times. This option can only be used with `-a`. Default is 1.
`-t` "*message-text*"	Message text. Alternatively, the message can be supplied as `stdin`. If the text has white space, use single or double quotes. Substitution is supported.
`-u` "*message-text*"	If no message-text is provided, `utwall` clears any message text that is in the `/etc/motd` file.
`-v`	Verbose mode.

OPERANDS

The following operands are supported:

`ALL`	Action is performed on all user having an Xsun process.
user`:`*display*	Action is performed on the given users (optional display number *display*) having an Xsun process.
`:`*display*	Action is performed on the users having display number *display*.

EXAMPLES

EXAMPLE 1 This command sends email to all users:

```
# /opt/SUNWut/sbin/utwall -m 'System policy change...' -t
'Tonight\nPlease log off' ALL
```

The email reads:

```
Subject: System policy change...
Tonight
Please log off
```

EXAMPLE 2 This command pops a window up on all sessions stating "Log off now!"

```
# /opt/SUNWut/sbin/utwall -d -t "Log off now!" ALL
```

EXAMPLE 3 This command pops a window up on jsmith's session on display 26 with the text from *messagefile*

```
# /opt/SUNWut/sbin/utwall -d jsmith:26 < messagefile
```

EXAMPLE 4 This command pops a window up with a greeting to the user on display number 10

```
# /opt/SUNWut/sbin/utwall -d -t "Hello" :10
```

ATTRIBUTES

See `attributes(5)` for descriptions of the following attributes:

Attribute Type	Attribute Value
Availability	SUNWuto

SEE ALSO

`wall(1M)`, `mailx(1M)`, `utaudio(1)`

NOTES

When Sun Ray appliances are configured for Xinerama, only the origin screen displays the `utwall` message.

NAME

`utxconfig` - Sun Ray appliance X server configuration utility.

SYNOPSIS

/opt/SUNWut/bin/utxconfig [-a] [-c *config-file*] [-d *display*] [-D] [-l] [-L *geom*] [-m *multihead*] [-p *pcolor*] [-r *res*] [-R *geom*] [-s *asize*] [-S *screen-order*] [-t *token*] [-x *xinerama*]

/opt/SUNWut/bin/utxconfig -e [-d *display*] [-t *token*]

/opt/SUNWut/bin/utxconfig [-o] [-f *file*]

/opt/SUNWut/bin/utxconfig [-i] [-f *file*]

DESCRIPTION

`utxconfig` displays and configures X server parameters for Sun Ray sessions. The changes to the X server are not evident until a restart of the X server process. For example, log out, then log in.

OPTIONS

The following options are supported.

`-a`	Allows the setting or listing of the default values. Only superuser may change the default settings.
`-c` *config-file*	Sets a specific *configfile* to use. The usage of this option is beyond the scope of this manual.
`-d` *display*	Sets the X display variable used to determine the Sun Ray appliance session. Otherwise, the DISPLAY environment variable is used. Users must have access to an X server attached to their session before they can read or change the settings for that session.
`-D`	Debug flag.
`-e`	Erases all specific settings for the session. All settings return to their default values.
`-f` *file*	Specifies a file to be used in conjunction with `-o` or `-i`.
`-i`	Populates the system settings database from a comma delimited text record such as the one produced by `-o`. Input is taken from the standard input unless `-f` is specified. You must be root to use this option.
`-l`	Lists out the current settings for the session. If no specific values have been set for the session, the default values are printed.
`-L` *geom*	Lists out the X server screen device start-up arguments for the user preferred geometry set with `-R` or for *geom* if none is set. The use of this option is beyond the scope of this manual.

`-m` *multihead*	Enables or disables multihead mode for X session startup. By default, if a session is started on a multihead terminal group, then the session starts in multihead mode to match the terminal group with an appropriate number of screens and geometry. Specify "off" to disable this behavior and the session starts on a single terminal with one screen.
`-o`	Output all system settings in a comma-delimited text record. Intended for use with `-i`. Outputs to standard output unless `-f` is specified.
`-p` *pcolor*	Parameter that specifies the level of support for the PseudoColor (8-bit) visual in the X server. The PseudoColor visual is not enabled by default. The accepted values for *pcolor* are "off", "on", and "default". A *pcolor* value of “off” will disable the PseudoColor visual. A *pcolor* value of "on" will enable the PseudoColor visual, but the TrueColor visual (24-bit) will remain the default. A *pcolor* value of "default" will enable the PseudoColor visual and make it the default visual, although the TrueColor visual will still be available.
`-r` *res*	Parameter that specifies a resolution (number of pixels) that the X server should provide for the session. The format of *res* is *WIDTH*x*HEIGHT*, for example 1280x1024. `utxconfig` enforces restrictions on the possible widths and heights that can be specified.
`-R` *geom*	Specify a preferred screen geometry in the form *COLS*x*ROWS*. At X server startup this geometry overrides the terminal group geometry on which the session is started. See `-m`.
`-s` *asize*	Enable or disable the selection of a resolution that best matches the resolution capabilities of the terminal on which the X session is started. This may be different (and overrides) the resolution set with `-r`. By default, the best selection is chosen. To disable this behavior, specify "off".
`-S` *screen-order*	Specify a preferred screen number order for the session’s screen group. The order must be a legal for standard Xsun (Sun Ray X server) screen placement.
`-t` *token*	Allows the setting of a specific token to use. The use of this option is beyond the scope of this manual.
`-x` *xinerama*	Enable or disable XINERAMA extension (not supported under the Solaris 2.6 operating environment; see man page for `Xserver(1)` on Solaris 8). By default, XINERAMA is disabled. To enable use of XINERAMA under Solaris 7 or 8 operating environment, specify "on". To revert to default behavior, specify "off".

EXAMPLES

EXAMPLE 1 This command enables PseudoColor visual on a 1024 x 768 screen:

```
% /opt/SUNWut/bin/utxconfig -r 1024x768 -p on
```

EXAMPLE 2 This command configures a multihead group with two screens, right and left:

```
% /opt/SUNWut/bin/utxconfig -m on -R 2x1 -S 0,1
```

ATTRIBUTES

See `attributes(5)` for descriptions of the following attributes:

Attribute Types	Attribute Values
Availability	SUNWuta
Interface Stability	Evolving

SEE ALSO

`Xserver(1)`

NOTES

The settings are actually maintained on the basis of an authentication token and do not remain specific to a single X display number.

NAME

`utxset` - Sun Ray appliance mouse configuration and screen blanking utility.

SYNOPSIS

/opt/SUNWut/lib/utxset [`-a` *accel*] [`-b` *blank*] [`-d`] [`-f` *file*] [`-t` *thresh*] [`-v`]

DESCRIPTION

The `utxset` command configures mouse acceleration and sets screen blanking characteristics of the Sun Ray appliances. It is executed internally by the X11 server to implement changes initiated by the `xset`(1) command.

OPTIONS

The following options are supported.

`-a` *accel* — Set the mouse acceleration to *accel* pixels/second^2. The acceleration is specified as an integer or floating point value.

`-b` *blank* — Set the Sun Ray Energy Star compliant monitor blanking interval to *blank* minutes. The blanking interval specifies how long the unit should wait (in minutes) after any user input before placing the monitor into an energy-saving standby mode. User input is defined as the moving the mouse or pressing a key. A *blank* value of zero disables the energy-saving mode.

`-d` — Run as a daemon process. With this option, `utxset` forks a copy of itself to run in the background which makes the settings changes, and waits for confirmation. The original command returns immediately.

`-f` *file* — Obtain the session ID from the *file*. Ordinarily, the session ID is obtained using the DISPLAY environment variable. With this option, the session ID is found on a line in *file* that begins with "`SESSION=`".

`-t` *thresh* — Set the mouse threshold to *thresh* pixels/second. The threshold velocity is specified as an integer or floating point value.

`-v` — Verbose mode. Use is beyond the scope of this document.

ATTRIBUTES

See `attributes(5)` for descriptions of the following attributes:

Attribute Type	Attribute Value
Availability	SUNWuto

SEE ALSO

`utsettings(1)`, `xset(1)`, `Xserver(1)`, `X11(1)`

NOTES

An option is required for this command.

A behavior of the mouse movement is that the cursor will accelerate at *accel* pixels/second^2 when a threshold velocity of *thresh* pixels/second is exceeded.