



# Sun Ray Server Software 2.0 Reference Manual

---

Sun Microsystems, Inc.  
4150 Network Circle  
Santa Clara, CA 95054 U.S.A.  
650-960-1300

Part No. 816-7399-10  
November 2002, Revision A

Send comments about this document to: [docfeedback@sun.com](mailto:docfeedback@sun.com)

Copyright 2002 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed at <http://www.sun.com/patents> and one or more additional patents or pending patent applications in the U.S. and in other countries.

This document and the product to which it pertains are distributed under licenses restricting their use, copying, distribution, and decompilation. No part of the product or of this document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any.

Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and in other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, AnswerBook2, docs.sun.com, and Solaris **[AND ATTRIBUTION OF ALL OTHER SUN TRADEMARKS MENTIONED SIGNIFICANTLY THROUGHOUT PRODUCT OR DOCUMENTATION. REMOVE THIS TEXT!]** are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and in other countries.

All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and in other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc. **[THIRD-PARTY TRADEMARKS THAT REQUIRE ATTRIBUTION APPEAR IN 'TMARK.' IF YOU BELIEVE A THIRD-PARTY MARK NOT APPEARING IN 'TMARK' SHOULD BE ATTRIBUTED, CONSULT YOUR EDITOR OR THE SUN TRADEMARK GROUP FOR GUIDANCE.]**

The OPEN LOOK and Sun™ Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

**[IF ENERGY STAR INFORMATION IS REQUIRED FOR YOUR PRODUCT, COPY THE ENERGY STAR GRAPHIC FROM THE REFERENCE PAGE AND PASTE IT HERE, USING THE "GraphicAnchor" PARAGRAPH TAG. ALSO, COPY THE ENERGY STAR LOGO TRADEMARK ATTRIBUTION FROM THE REFERENCE PAGE AND PASTE IT ABOVE WHERE THIRD-PARTY TRADEMARKS ARE ATTRIBUTED. (ENGLISH COPYRIGHT ONLY). DELETE THIS TEXT.]**

Use, duplication, or disclosure by the U.S. Government is subject to restrictions set forth in the Sun Microsystems, Inc. license agreements and as provided in DFARS 227.7202-1(a) and 227.7202-3(a) (1995), DFARS 252.227-7013(c)(1)(ii) (Oct. 1998), FAR 12.212(a) (1995), FAR 52.227-19, or FAR 52.227-14 (ALT III), as applicable.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

---

Copyright 2002 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, Etats-Unis. Tous droits réservés.

Sun Microsystems, Inc. a les droits de propriété intellectuels relatants à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et sans la limitation, ces droits de propriété intellectuels peuvent inclure un ou plus des brevets américains énumérés à <http://www.sun.com/patents> et un ou les brevets plus supplémentaires ou les applications de brevet en attente dans les Etats-Unis et dans les autres pays.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a.

Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit pourront être dérivées des systèmes Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, AnswerBook2, docs.sun.com, et Solaris **[AND ATTRIBUTION OF ALL OTHER SUN TRADEMARKS MENTIONED SIGNIFICANTLY THROUGHOUT PRODUCT OR DOCUMENTATION. REMOVE THIS TEXT!]** sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays.

Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc. **[THIRD-PARTY TRADEMARKS THAT REQUIRE ATTRIBUTION APPEAR IN 'TMARK.' IF YOU BELIEVE A THIRD-PARTY MARK NOT APPEARING IN 'TMARK' SHOULD BE ATTRIBUTED, CONSULT YOUR EDITOR OR THE SUN TRADEMARK GROUP FOR GUIDANCE.]**

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciées de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.



LA DOCUMENTATION EST FOURNIE "EN L'ÉTAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFAÇON.



# Contents

---

**Preface 1**

auth.props 5

sunray 11

utaction 13

utadem 15

utadm 17

utadmin.conf 21

utaudio 23

utauthd 27

utcapture 29

utcard 33

utconfig 35

utcrypto 37

utdesktop 39

utdetach 43

utdevmgrd 45

utdssync 49

utfwadm 51

utfwsync 55

utglpolicy 57

utgroupsig 61

utgstatus 63

utinstall 67

utkiosk 69

utmhadm 71

utmhconfig 73

utmhscreen 75

utparallel, utserial 77

utparalleld 79

utpolicy 81

utpreserve 87

utpw 89

utquery 91

utrcmd 93

utreader 97

utreplica 99

utresadm 101

utresdef 103

utrestart 105

utselect 107

utserial, utparallel 109

utseriald 111

utsession 113

utsessiond 117

utset 119

utsettings 123

utslaunch 125

utslaunch.properties 127

utsumc 129

utswitch 131

utuser 133

utwall 137

utxconfig 139

utxlock 143

utxset 145



# 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 <code>errno</code> 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: <ul style="list-style-type: none"> <li>Commands</li> <li>Modifiers</li> <li>Variables</li> <li>Expressions</li> <li>Input Grammar</li> </ul>
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 <code>example%</code> or if the user must be superuser, <code>example#</code> . 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 <code>attributes(5)</code> 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.

<b>NAME</b>	auth.props - Sun Ray authentication daemon configuration file.
<b>SYNOPSIS</b>	/etc/opt/SUNWut/auth.props
<b>DESCRIPTION</b>	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.
<b>OPTIONS</b>	The following options are supported:
	<p><code>acceptRedirectToken=<i>boolean</i></code> 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 <code>false</code> causes the target Authentication Manager to distrust the authenticated token and re-authentication is required.</p> <p>The default is set to <code>false</code> for higher security. Set to <code>true</code> if you want to accept authenticated redirects.</p>
	<p><code>adminConfigFile=<i>filename</i></code> This file contains the administrative database configuration information.</p>
	<p><code>allowAnnotations=<i>boolean</i></code> 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.</p>
	<p><code>allowFWLoad=<i>boolean</i></code> Specifies whether or not the <code>utload</code> command is allowed to download firmware to appliances connected to this Authentication Manager.</p>
	<p><code>allowLANConnections=<i>boolean</i></code> When set true the appliance connections are allowed from <code>localhost</code> as well as from non-Sun Ray interconnect interfaces.</p>
	<p><code>cbport=<i>portNumber</i></code> UNSUPPORTED. The Authentication Manager listens on this port for connections from the <code>utsessiond</code> daemon and other programs, such as <code>utload</code>.</p>
	<p><code>cbtimeout=<i>seconds</i></code> UNSUPPORTED. Specifies the read timeout in seconds for programs that connect to the <code>cbport</code>.</p>

<code>controllers=<i>maximum</i></code>	UNSUPPORTED. Specifies the maximum number of spare threads that are available for handling new connections from applications such as <code>utload(1M)</code> .
<code>enableGroupManager=<i>boolean</i></code>	UNSUPPORTED. Flag to turn on the group manager function.
<code>enableLoadBalancing=<i>boolean</i></code>	Flag to turn on group manager load balancing.
<code>enableMulticast=<i>boolean</i></code>	UNSUPPORTED. Flag to enable/disable use of multicast in group manager. If disabled, group manager will use broadcast.
<code>forceSessionLocation=<i>boolean</i></code>	UNSUPPORTED. Flag to force use of <code>sessionHost</code> and <code>sessionPort</code> settings from this file regardless of the various authentication modules.
<code>gmDebug=<i>level</i></code>	UNSUPPORTED. Group manager debugging level.
<code>gmKeepAliveInterval=<i>seconds</i></code>	UNSUPPORTED. The group manager uses this as the time in seconds between broadcast keepalive messages.
<code>gmport=<i>port</i></code>	UNSUPPORTED. The group manager uses this port to send and receive keepalive/discovery messages from other Authentication Managers.
<code>gmSignatureFile=<i>file</i></code>	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.
<code>log=<i>filename</i></code>	UNSUPPORTED. This option specifies a file that contains the log messages.
<code>logAddTimeStamp=<i>boolean</i></code>	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.

<code>logFacility=<i>value</i></code>	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
<code>Log files</code>	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:  <code>logPriClientError=<i>value</i></code>  <code>logPriDebug=<i>value</i></code>  <code>logPriNotice=<i>value</i></code>  <code>logPriWarning=<i>value</i></code>  <code>logPriConfigError=<i>value</i></code>  <code>logPriUnexpectedError=<i>value</i></code>
<code>maxStarting=<i>maximum</i></code>	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.
<code>moduleDif=<i>directorName</i></code>	UNSUPPORTED. Specifies the location of the authentication modules.
<code>multicastTTL=<i>integer</i></code>	UNSUPPORTED. Time-to-live parameter for forwarding multicast packets. If set above one, keepalive messages can pass through routers.
<code>noClaimSleepTime=<i>seconds</i></code>	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.
<code>policy=<i>filename</i></code>	Specifies the location of the authentication policy specification.
<code>port=<i>portNumber</i></code>	The utauthd daemon listens on this port for connections from Sun Ray appliances.

<code>remoteSelect=<i>boolean</i></code>	If true, the remote server selection option of the <code>utselect(1)</code> command is enabled by default.
<code>reportAllDesktopEvents=<i>boolean</i></code> <code>n</code>	UNSUPPORTED. When true, all desktop events are reported instead of being filtered to just those events that change the "exists" state of the appliance.
<code>restrictSunrayIfs=<i>boolean</i></code>	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.
<code>selectAtLogin=<i>boolean</i></code>	If true, activates a <code>utselect -L</code> 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 <code>utselect</code> man page for more information on the <code>-L</code> option.
<code>sessionHost=<i>hostname</i></code>	UNSUPPORTED. Specifies the host name of the server that is running the default <code>utsessiond</code> for this Authentication Manager.
<code>sessionPort=<i>portNumber</i></code>	UNSUPPORTED. Specifies the port number of the server that is running the default <code>utsessiond</code> for this Authentication Manager.
<code>sessionTypesFile=<i>filename</i></code>	Specifies a file that contains mappings from session types to the associated session startup and shutdown commands.
<code>smttimeout=<i>seconds</i></code>	UNSUPPORTED. Specifies the read timeout in seconds for reading messages from the <code>utsessiond</code> daemon.
<code>termAddrIsSecret=<i>boolean</i></code>	UNSUPPORTED. When true, the IP address and port of appliances are not reported in the dynamic status information provided on port <code>cbport</code> in response to the string.
<code>terminateEnable=<i>boolean</i></code>	UNSUPPORTED. Enables experimental code in <code>utauthd</code> .
<code>timeout=<i>seconds</i></code>	UNSUPPORTED. Appliances are required to send a message to the Authentication Manager at least once every time period specified by <i>seconds</i> .



`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.

`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/lib/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)



<b>NAME</b>	<code>utaction</code> - Sun Ray appliance connect/disconnect action				
<b>SYNOPSIS</b>	<code>/opt/SUNWut/lib/utaction [-c <i>ccmd</i>] [-d <i>dcmd</i>] [-D <i>display</i>] [-i] [-t <i>sec</i>]</code>				
<b>DESCRIPTION</b>	The <code>utaction</code> program provides a way to execute commands when a Sun Ray appliance session is connected or disconnected. The <code>ccmd</code> is invoked using <code>sh(1)</code> whenever the session is connected to an appliance. Similarly, the <code>dcmd</code> is invoked using <code>sh(1)</code> whenever the session is disconnected from an appliance. Normally, action is not taken on the initial state of the session (when <code>utaction</code> is first run) unless the <code>-i</code> option is used.				
<b>OPTIONS</b>	The following options are supported. <ul style="list-style-type: none"> <li><code>-c <i>ccmd</i></code> Run this command when the current session is connected to an appliance.</li> <li><code>-d <i>dcmd</i></code> Run this command when the current session is disconnected to an appliance.</li> <li><code>-D <i>display</i></code> This option will set the X display variable that is to be used in determining the Sun Ray enterprise appliance session. Otherwise the <code>DISPLAY</code> environment variable is used.</li> <li><code>-i</code> Run the connect or disconnect command immediately, whichever is appropriate.</li> <li><code>-t <i>sec</i></code> This option specifies a time-delay in seconds for the actions. In that case, the <code>ccmd</code> or <code>dcmd</code> will not be invoked unless the session remains in the connected or disconnected state, respectively, for at least <code>sec</code> seconds.</li> </ul>				
<b>EXAMPLES</b>	<p><b>EXAMPLE 1</b> This command invokes the CDE screen lock whenever the session is disconnected</p> <pre>% utaction -d '/usr/dt/bin/dtaction LockDisplay' &amp;</pre>				
<b>ATTRIBUTES</b>	See <code>attributes(5)</code> for descriptions of the following attributes:				
	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Attribute Type</th> <th style="text-align: center;">Attribute Value</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">Availability</td> <td style="text-align: center;">SUNWuto</td> </tr> </tbody> </table>	Attribute Type	Attribute Value	Availability	SUNWuto
Attribute Type	Attribute Value				
Availability	SUNWuto				
<b>NOTES</b>	The <code>ccmd</code> and <code>dcmd</code> are each only one argument to <code>utaction</code> . Quotes should be used if a command contains multiple words.				



<b>NAME</b>	utadem - Sun Ray audio driver emulator.
<b>SYNOPSIS</b>	/dev/utadem
<b>DESCRIPTION</b>	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.
<b>API</b>	<p>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 <code>-d device</code> 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:</p> <p><b>Audio Data Formats</b>                      The data formats supported depend on the daemon. Please refer to the daemon documentation for its capabilities.</p> <p><b>Audio Ports</b>                                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 <code>record.avail_ports</code> and <code>play.avail_ports</code> fields of the <code>audio_info</code> structure. Although the ports can be controlled directly, the actual audio output is generally a mix of multiple services, so the <code>play.gain</code> setting is the contribution of this audio device to the total experience. Since recording is exclusive of a single service, the <code>record.gain</code> and <code>record.balance</code> controls directly affect the hardware gain.</p> <p><b>Sample Granularity</b>                      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 <code>play.samples</code> and <code>record.samples</code> fields of the <code>audio_info</code> structure.</p>

**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)`



<b>NAME</b>	<code>utadm</code> - Sun Ray network and DHCP configuration utility.
<b>SYNOPSIS</b>	<code>/opt/SUNWut/sbin/utadm -a <i>interface-name</i></code> <code>/opt/SUNWut/sbin/utadm -c</code> <code>/opt/SUNWut/sbin/utadm -d <i>interface-name</i></code> <code>/opt/SUNWut/sbin/utadm -f</code> <code>/opt/SUNWut/sbin/utadm -l</code> <code>/opt/SUNWut/sbin/utadm -n</code> <code>/opt/SUNWut/sbin/utadm -p</code> <code>/opt/SUNWut/sbin/utadm -r</code> <code>/opt/SUNWut/sbin/utadm -A <i>subnetwork</i></code> <code>/opt/SUNWut/sbin/utadm -D <i>subnetwork</i></code> <code>/opt/SUNWut/sbin/utadm -L on   off</code>
<b>DESCRIPTION</b>	The <code>utadm</code> 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: <code>-a</code> , <code>-c</code> , <code>-d</code> , <code>-f</code> , <code>-l</code> , <code>-n</code> , <code>-p</code> , <code>-r</code> , <code>-A</code> , or <code>-D</code> . 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 <i>hostname-interface-name</i>
/etc/networks	SunRay- <i>interface-name</i> 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.

- l print the current configuration for all the Sun Ray subnetworks. This includes remote subnetworks.
- 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. If LAN connection support is on, will prompt the user if it should be turned off. Default is yes (turn it off).
- A configure the subnetwork specified as a Sun Ray subnetwork. This option only configures the DHCP service to allocate IP address and/or to provide Sun Ray parameters to Sun ray clients. It also will automatically turn on support for LAN connections from a shared subnetwork.
- D delete the subnetwork specified form the list of configured Sun Ray subnetworks.
- L on| will turn on or off support for LAN connections from a shared  
off subnetwork.

**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/opt/SUNWut/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 7012.  Example: 7012

- `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 Administrator's Guide*

<b>NAME</b>	utaudio - Sun Ray audio services connection utility.
<b>SYNOPSIS</b>	<b>/opt/SUNWut/bin/utaudio</b>
<b>    csh</b>	<b>setenv AUDIODEV 'utaudio'</b>
<b>    ksh</b>	<b>export AUDIODEV='utaudio'</b>
<b>    sh</b>	<b>AUDIODEV='utaudio';export AUDIODEV</b>
<b>DESCRIPTION</b>	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.
<b>OPTIONS</b>	There are no options for utaudio.
<b>API</b>	<p>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."</p> <p>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.</p> <p><b>Audio Data Formats</b>            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.</p>

**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/nctl`  
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)`, `prioctl(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.



<b>NAME</b>	utauthd - Sun Ray appliance authentication daemon.				
<b>SYNOPSIS</b>	<code>/opt/SUNWut/lib/utauthd -b   -e</code>				
<b>DESCRIPTION</b>	The <code>utauthd</code> 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.				
<b>OPTIONS</b>	<p>The following options are supported.</p> <p><code>-b</code>           Begin execution of the daemon.</p> <p><code>-e</code>           End execution of the daemon.</p> <p>Without arguments, the default is <code>-b</code>.</p>				
<b>FILES</b>	<p>The following files are used by the daemon:</p> <ul style="list-style-type: none"> <li>■ <code>/etc/init.d/utsvc</code> This is the system startup script that invokes the daemon. <code>/opt/SUNWut/utsessiond</code>, the Session Manager, performs the actual session switching function.</li> <li>■ <code>/etc/opt/SUNWut/auth.props</code> The Authentication Manager's configuration file.</li> <li>■ <code>/etc/opt/SUNWut/policy/utpolicy</code> This file determines what policy is used by the Sun Ray server.</li> </ul> <p>The authentication manger is normally started by running <code>utsvc</code> with the <code>start</code> or <code>restart</code> argument. The <code>start</code> argument starts both the session manager and the authentication manager, so all of the sessions are lost. The <code>restart</code> argument only starts the authentication manager, so all the sessions are continued.</p>				
<b>ATTRIBUTES</b>	See <code>attributes(5)</code> for descriptions of the following attributes:				
	<table border="1"> <thead> <tr> <th>Attribute Type</th> <th>Attribute Value</th> </tr> </thead> <tbody> <tr> <td>Availability</td> <td>SUNWuto</td> </tr> </tbody> </table>	Attribute Type	Attribute Value	Availability	SUNWuto
Attribute Type	Attribute Value				
Availability	SUNWuto				
<b>SEE ALSO</b>	<code>auth.props(4)</code> , <code>utpolicy(1M)</code>				



<b>NAME</b>	<code>utcapture</code> - Capture packet information from the Authentication Manager.
<b>SYNOPSIS</b>	<pre> /opt/SUNWut/sbin/utcapture -h /opt/SUNWut/sbin/utcapture [-r] [-s server] [desktopID1 desktopID2 ...] /opt/SUNWut/sbin/utcapture -i filename </pre>
<b>DESCRIPTION</b>	<p>The <code>utcapture</code> command connects to the Authentication Manager and monitors latency, packets sent, and packets dropped between the Sun Ray server and the Sun Ray appliances.</p> <p><code>utcapture</code> writes the captured information to <code>stdout</code> in the following format:</p> <pre> TERMINALID  TIMESTAMP  TOTAL  PACKET  TOTAL  LOSS  BYTES  SENT  PERCENT LOSS  LATENCY </pre>
<b>OPTIONS</b>	<p>The following options are supported.</p> <p><code>-h</code>            Help for using the command.</p> <p><code>-i filename</code>    Use an input file to search for Sun Ray appliances that experienced dropped packets. A file is created using <code>utcapture</code>:</p> <pre> # /opt/SUNWut/sbin/utcapture -r &gt; /tmp/filename </pre> <p>The process is allowed to run for several minutes or hours. The <code>utcapture</code> command is used again:</p> <pre> # /opt/SUNWut/sbin/utcapture -i /tmp/filename </pre> <p>The output is only the appliances that experienced dropped packets.</p> <p><code>-r</code>            Write captured data to <code>stdout</code> every 15 seconds in raw, continuous format.</p> <p><code>-s server</code>     Specify the Sun Ray <i>server</i> from which to capture data. If outside of the domain of the host running <code>utcapture</code>, the Sun Ray server hostname must be fully qualified. By default, the server monitored is the host running <code>utcapture</code>.</p> <p>When no option is specified, <code>utcapture</code> writes to <code>stdout</code> at 15 second intervals if there is any change in packet loss or latency exceeds 10 ms for any Sun Ray appliance.</p>

**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 -s 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.

`utcapture` does not report latency for Sun Ray appliances using firmware versions of 1.3 or less.

When using `-r` 0.000 will listed for PERCENT LOSS as 0.000 for every interval that has no loss. When not using `-r` this column will be blank. If the output is to be processed by commands that are column oriented, you must use `-r`. This includes using it as input to `utcapture -i`.





<b>NAME</b>	utcard - Sun Ray server smart card configuration utility
<b>SYNOPSIS</b>	<pre> /opt/SUNWut/sbin/utcard -a <i>filename</i> /opt/SUNWut/sbin/utcard -d <i>name,version</i> /opt/SUNWut/sbin/utcard -h /opt/SUNWut/sbin/utcard -l /opt/SUNWut/sbin/utcard -p <i>name,version</i> /opt/SUNWut/sbin/utcard -r <i>name,version,new-position</i> /opt/SUNWut/sbin/utcard -u </pre>
<b>DESCRIPTION</b>	<p>The <code>utcard</code> command allows configuration of different types of smartcards in the Sun Ray administration database.</p> <p>The administrator must first place a configuration file for a specific smartcard in the <code>/etc/opt/SUNWut/smartcard</code> directory. This file must have a <code>.cfg</code> extension. The smartcard definition in the <code>.cfg</code> file is added to the LDAP datastore by using the <code>-a</code> 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 <code>-r</code> option.</p>
<b>OPTIONS</b>	<p>The following options are supported.</p> <pre> -a <i>filename</i>      Add the card specified within <i>filename</i> that is in the                   /etc/opt/SUNWut/smartcard directory -d                Delete the card specified with <i>name, version</i>. -h                Show usage information -l                List all configured cards -p                Show the standard properties for the card specified with <i>name,                   version</i>. -r                Reorder the card specified with <i>name, version, to new-position</i>. -u                List unconfigured cards available for configuration as                   determined by the <code>.cfg</code> files in <code>/etc/opt/SUNWut/                   smartcard</code> </pre>
<b>USAGE</b>	Use this command only on a Sun Ray server that has been configured for administration by the <code>utconfig</code> 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)`



<b>NAME</b>	<code>utcrypto</code> - Sun Ray privacy administration utility.
<b>SYNOPSIS</b>	<pre> /opt/SUNWut/sbin/utcrypto /opt/SUNWut/sbin/utcrypto -a key=value ... /opt/SUNWut/sbin/utcrypto -d /opt/SUNWut/sbin/utcrypto -e [-f filename] /opt/SUNWut/sbin/utcrypto -h /opt/SUNWut/sbin/utcrypto -m key=value ... /opt/SUNWut/sbin/utcrypto -o [-f filename] </pre>
<b>DESCRIPTION</b>	<p>The <code>utcrypto</code> command allows the administrator to configure privacy options for the Sun Ray server. These settings include upstream/downstream encryption and upstream/downstream authentication. These settings will affect all sessions.</p> <p><code>utcrypto</code> operations that only display information may be run by any user. Operations that change or delete data must be run as superuser.</p>
<b>OPTIONS</b>	<p>The following options are supported.</p> <ul style="list-style-type: none"> <li>-a Add the privacy settings for all sessions. The <code>-a</code> option must be followed by a series of <code>key=value</code> pairs separated by spaces. Valid <code>key=value</code> pairs are described below. If a <code>value</code> pair is not passed in, the value for that key will be set to a default value:</li> <li>-d Delete the privacy settings for all sessions.</li> <li>-e Take the privacy settings in comma-delimited format from <code>stdin</code> and add/modify the settings for all sessions. This will be a space-delimited list of <code>key=value</code> pairs. If followed by the <code>-f</code> option and a <code>filename</code>, it will read the settings from a file.</li> <li>-f Specifies a filename for the <code>-e</code> or <code>-o</code> option.</li> <li>-h Displays the usage message.</li> <li>-m Modify a privacy settings for all sessions. The <code>-m</code> option must be followed by a series of <code>key=value</code> pairs separated by spaces. Valid <code>key=value</code> pairs are described below. If a <code>value</code> is not passed in, the value for that key will remain unchanged.</li> <li>-o Dump all privacy settings in comma-delimited format to <code>stdout</code>. If followed by the <code>-f</code> option and a <code>filename</code>, the settings will be dumped to the file. If the file with the specified by <code>filename</code> exists, then a warning message will be generated and the script will exit.</li> </ul>

The following are valid *key=value* pairs:

```
enc_up_type   ARCFOUR | none | default
enc_down_type ARCFOUR | none | default
auth_up_type  none    | default
auth_down_type simple | none | default
e
mode          hard   | soft | default
```

The keyword default for all keys is set to the value of the default configuration, if it exists. If there a default configuration has not been set, then the value is set to hard-coded defaults. The hard-coded default values for the first four keys are none and soft for the mode key.

**EXAMPLES**

**EXAMPLE 1** This adds the configuration for upstream ARCfour encryption and simple downstream authentication. Note - that since `enc_down_type` and `auth_down_type` are not specified, they will take on the default values

```
# /opt/SUNWut/sbin/utcrypto -a enc_up_type=ARCFOUR
auth_down_type=simple
```

**EXAMPLE 2** This command modifies the default configuration. Upstream encryption is turned off and downstream encryption is set to ARCfour

```
/opt/SUNWut/sbin/utcrypto -m enc_up_type=none
enc_down_type=ARCFOUR
```

**EXAMPLE 3** This command removes the default configuration.

```
/opt/SUNWut/sbin/utcrypto -d
```

**ATTRIBUTES**

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

Attribute Type	Attribute Value
Availability	SUNWuto

<b>NAME</b>	utdesktop - Sun Ray appliance administration utility.
<b>SYNOPSIS</b>	<pre> /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 </pre>
<b>DESCRIPTION</b>	<p>The <code>utdesktop</code> command allows the user to manage Sun Ray appliances connected to the Sun Ray server the command is run on. The information that <code>utdesktop</code> 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.</p> <p><code>utdesktop</code> operations that only display information may be run by any user. Operations that add, edit, or delete data must be run by the superuser.</p>
<b>OPTIONS</b>	<p>The following options are supported.</p> <ul style="list-style-type: none"> <li>-a                    Add appliance with the specified desktop-ID, location, and <i>desktopID,location,other</i> other information properties. Note that the 3 comma-delimited values should be enclosed within quotes. You must be root to use this option.</li> <li>-a -f <i>filename</i>    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.</li> <li>-d <i>desktopID</i>      Delete the appliance with the specified desktop-ID. You must be root to use this option.</li> </ul>

- d -f *filename* Batch delete multiple appliances using input from 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.



<b>NAME</b>	utdetach - Detach the current session from the Sun Ray appliance.				
<b>SYNOPSIS</b>	<b>/opt/SUNWut/bin/utdetach</b>				
<b>DESCRIPTION</b>	<p>The <code>utdetach</code> 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.</p> <p>This command is primarily executed by users of the non-smart card mobility feature so as to disconnect their "mobile" sessions.</p> <p>The Sun Ray server starts an instance of <code>utslaunch (1M)</code> for each session whenever a user logs into a Sun Ray appliance via <code>dtlogin</code>. This makes the <code>utdetach</code> command available to users as a hotkey sequence. The default hotkey sequence is Shift + Pause and can be configured in the <code>utslaunch.properties</code> file.</p>				
<b>OPTIONS</b>	There are no options for <code>utdetach</code> .				
<b>EXAMPLES</b>	<p><b>EXAMPLE 1</b> This command disconnects the current session from the appliance the user is currently using.</p> <pre>% utdetach</pre>				
<b>FILES</b>	<p>The following files are used:</p> <ul style="list-style-type: none"> <li>■ <code>/etc/opt/SUNWut/utslaunch_defaults.properties</code> site-wide defaults</li> <li>■ <code>~/.utslaunch.properties</code> user's defaults</li> <li>■ <code>/etc/opt/SUNWut/utslaunch_mandatory.properties</code> site-wide mandatory defaults</li> </ul>				
<b>EXIT STATUS</b>	<p>The following exit values are returned:</p> <table border="0"> <tr> <td style="padding-right: 20px;">0</td> <td>Success</td> </tr> <tr> <td>1</td> <td>Failure</td> </tr> </table>	0	Success	1	Failure
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)`

<b>NAME</b>	utdevmgrp - Sun Ray device manager daemon.
<b>SYNOPSIS</b>	<code>/opt/SUNWut/lib/utdevmgrp [-a <i>authlist</i>] [-c <i>authfile</i>] [-d] [-k <i>authprops</i>] [-o <i>optroot</i>] [-p <i>port</i>] [-r] [-s <i>sigfile</i>] [-t]</code>
<b>DESCRIPTION</b>	<p>The <code>utdevmgrp</code> 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.</p> <p>If either the <code>-a</code> or the <code>-c</code> 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 <code>authlist</code> or <code>authfile</code> 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.</p> <p>The <code>optroot</code> directory (default <code>/tmp/SUNWut</code>) 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 <code>sessions</code> and <code>units</code> subdirectories.</p> <p>The <code>units</code> 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 <code>dev</code> and <code>devices</code> directories that list logical names for devices and geographically hierarchical names for devices.</p> <p>The <code>sessions</code> directory contains symbolic links into the <code>devices</code> 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 <code>DISPLAY</code> environment variable can then be used to find the devices on the 'current' appliance. The user's <code>UTDEVROOT</code> environment variable achieves this, and can be used to find devices that are 'currently' accessible. The <code>optroot</code> directory also includes the named pipe with which the device manager communicates to device driver services and the <code>session_info</code> directory, which contains user information important to internal workings of the device manager.</p> <p>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.</p> <p>Normally, the device manager finds the group signature file by looking into the authentication manager's configuration file (<code>/etc/opt/SUNWut/auth.props</code>),</p>

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.

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

## ENVIRONMENT VARIABLES

The following environment variables are used:

<code>DISPLAY</code>	Use to get the default X-Windows display number from within the user's environment.
<code>UTDEVROOT</code>	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** utdssync - Converts and synchronizes Sun Ray datastore service port.

**SYNOPSIS** /opt/SUNWut/sbin/utdssync [-v]

**DESCRIPTION** The `utdssync` command converts the Sun Ray datastore service port on the primary datastore server to the new default port 7012. It also synchronizes all the secondary servers to use the same port number.

`utdssync` is intended for use after software upgrades from Sun Ray Server Software 1.x to 2.0 have been completed on all the servers within the data sharing group.

**Note** – The command must be run with superuser privileges on the primary datastore server.

**OPTIONS** The following option is supported.

-v                    verbose mode

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

Attribute Type	Attribute Value
Availability	SUNWuta

**SEE ALSO** `utreplica(1M)`, `utrcmd(1M)`, `utinstall(1M)`



<b>NAME</b>	utfwadm - Sun Ray appliance firmware version management.
<b>SYNOPSIS</b>	<pre> /opt/SUNWut/sbin/utfwadm -A {-a   -e enetAddr} [-f firmware] {-n intf   -N subnetwork} ...  /opt/SUNWut/sbin/utfwadm -D {-a   -e enetAddr} [-f firmware] {-n intf   -N subnetwork} ...  /opt/SUNWut/sbin/utfwadm -P  /opt/SUNWut/sbin/utfwadm -R </pre>
<b>DESCRIPTION</b>	<p>The <code>utfwadm</code> 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.</p> <p>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.</p> <p>The <code>utfwadm</code> 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. <code>utfwadm</code> 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.</p> <p>It is possible to determine the firmware versions that are available and in use:</p> <ul style="list-style-type: none"> <li>■ In the <code>/tftpboot</code> directory are the firmware files. To verify the version of a firmware file, type: <pre> /opt/SUNWut/sbin/lzd &lt; /tftpboot/<i>firmware-filename</i>   what </pre> </li> <li>■ To identify the version of firmware that a particular Sun Ray appliance is using, type: <pre> /opt/SUNWut/sbin/utdesktop -p <i>desktopID</i> </pre> <p>Where <i>desktopID</i> is the full MAC address. This is displayed by pressing the three audio keys on the Sun Ray appliance keyboard simultaneously.</p> </li> </ul> <p>The Sun Ray subnetworks must have been previously set up using the <code>utadm(1M)</code> command. The <code>utfwadm</code> command is run under super-user privileges.</p>
<b>OPTIONS</b>	<p>The following options are supported.</p> <p><code>-A</code>                    This causes the given operation to be applied to all units attached to the given interfaces.</p>

- 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 *intf* Apply the given operation to units connected to the Ethernet interface *intf*. 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.
- N *subnetwork* apply the given operation to units attached on the specified subnetwork. Multiple subnetworks may be given. It can also specify the special keyword `all` to apply the operation to all configured Sun Ray subnetworks.

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)`





<b>NAME</b>	utglpolicy - View or set policy for entire failover group.
<b>SYNOPSIS</b>	<pre> /opt/SUNWut/sbin/utglpolicy -a [-g] [-k type] [-m] [-M] [-p] [-r type] [-s type] [-z type] [-S smartcard_type]  /opt/SUNWut/sbin/utglpolicy -a [-t {clear   add:terminalID   del:terminalID}]  /opt/SUNWut/sbin/utglpolicy -h  /opt/SUNWut/sbin/utglpolicy </pre>
<b>DESCRIPTION</b>	<p>The <code>utglpolicy</code> command returns or sets the <code>utpolicy</code> options which are used for the entire failover group. The <code>utglpolicy</code> command provides a command-line alternative to the Sun Ray Administration Tool which is normally used for setting group policies.</p> <p>You must be <code>root</code> to use this command.</p>
<b>OPTIONS</b>	<p>The following options is supported.</p> <p>Two categories of options are supported: Policy Setting and Card Reader Assignment</p> <p><code>-a</code> 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.</p> <p><b>POLICY SETTING</b></p> <p>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</p> <p><code>-g</code> Turn on session selection within a server group. Allows the user to select on which server the user's session is run.</p> <p><code>-k {card pseudo both}</code> Enables Controlled Access Mode (CAM) for the specified session type. Selecting <code>card</code> enables CAM for <code>card</code> sessions, <code>pseudo</code> for terminal sessions, and <code>both</code> enables CAM for all types of sessions. For functionality, at least one of the <code>-r</code>, <code>-s</code>, or <code>-z</code> options must be invoked with the same argument as the <code>-k</code> option. The <code>-k</code> option is not considered until the <code>utconfig</code> application has configured for Controlled Access Mode.</p>

- 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.
- S *smartcard\_type* Specify the smartcard type that should cause the Smartcard Login GUI (utsclogin) to be invoked when a card of the type in the list is inserted into the DTU. This is the smartcard token type (without the token ID) as provided by the Sun Ray software. If more than one smartcard type is to be specified, multiple -S options must be specified.  
  
**Note** – in order for the Smartcard Login GUI to operate properly, the /etc/pam.conf file must have at least one PAM module that supports PAM authentication configured for the utsclogin PAM client and using the PAM client name "utsclogin-<smartcard\_type>". utglpolicy does not configure nor check for proper configuration of the /etc/pam.conf file.

#### CARD READER ASSIGNMENT

**Note** – The Card Reader Assignment operation through the `-i` to `utglpolicy` has been deprecated in 2.0. The new `utreader` command is the appropriate tool to perform this operation.

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, `utgpolicy` does not support the `-i {clear |soft}` and `-t list` options.

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

With no options, the `utgpolicy` 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/utgpolicy -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/utgpolicy -k card -r card -z pseudo
```

**EXAMPLE 3** This command allows all appliances to be used with or without a smart card. Non-smart card mobile sessions are enabled. The smartcard types "JavaBadge" and "Payflex" cause the Smartcard Login GUI (`utslogin`) to be invoked when a card of these types is inserted into the DTU.

```
# /opt/SUNWut/sbin/utgpolicy -a -M -z both -S
JavaBadge -S Payflex
```

**ATTRIBUTES**

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

Attribute Type	Attribute Value
Availability	SUNWuta

**SEE ALSO**

`utpolicy(1M)`, `utreader(1M)`, `pam(3PAM)`

**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.

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 600 (read-write 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.

**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)



<b>NAME</b>	utgstatus - Display failover group status.
<b>SYNOPSIS</b>	<code>/opt/SUNWut/sbin/utgstatus [-s <i>hostname</i>]</code>
<b>DESCRIPTION</b>	<p>The <code>utgstatus</code> 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.</p> <p><code>utgstatus</code> displays information only and so can be run by any user.</p>
<b>OPTIONS</b>	<p>The following option is supported.</p> <p><code>-s <i>hostname</i></code>    Display all the failover group status information for the specified <i>hostname</i>.</p>
<b>EXAMPLES</b>	<p><b>EXAMPLE 1</b>    This command displays the failover group status for the local Sun Ray server</p> <pre>% /opt/SUNWut/sbin/utgstatus</pre> <p><b>EXAMPLE 2</b>    This command displays the failover group status for the server <code>sunray3</code>:</p> <pre>% /opt/SUNWut/sbin/utgstatus -s sun3</pre> <p>Information returned from this command looks similar to the following (to view this correctly, make the terminal window very wide):</p> <pre> host      flags      interface      flags           192.24.0.0/24 ----- sunray3   TN         192.24.0.136  UAM sunray1   T-        192.24.0.93   UA- sunray2   TN         192.24.0.95   UAM sunray-sras TN        192.24.0.96   U-- </pre> <p>for a typical LAN-based configuration, or:</p> <pre> host  flags      interface      flags      interface      flags      interface      flags           193.25.0.0/24      10.21.0.0/16      10.22.0.0/16 ----- sunray-01 TN    193.25.0.10   U--    10.21.0.1     UAM    10.22.0.1     UAM sunray-24 TN    193.25.0.83   U--    10.21.0.24    UAM    10.22.0.24    UAM sunray-06 TN    193.25.0.12   U--    10.21.0.6     UAM    10.22.0.6     UAM sunray-02 TN    193.25.0.8    U--    10.21.0.2     UAM    10.22.0.2     UAM sunray-16 TN    193.25.0.20   U--    10.21.0.16    UAM    10.22.0.16    UAM </pre> <p>for a typical interconnect-based configuration.</p>

### Explanation of utgstatus information:

The Network/Netmask values are denoted in CIDR (Classless Inter Domain Routing) network address notation, where the initial value (192.24.0.0) 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.

### Host Status Flags

- T Trusted — The trusted hosts are members of this failover group because they share the same group signature.
- N oNline — The server is configured to participate in load balancing (see the utadm man page for a description of the -n option).

### Interface Status Flags

- U Up — The interface is currently reachable by this host.
- A Available — The interface is available for Sun Rays to connect to it and get service.
- M Managing - The interface is configured to manage Sun Rays on its local subnet (in other words, utadm -a was run to configure the interface for Sun Ray service).

In the first example above, all hosts are part of the same failover group. All hosts but sunray1 are "online", which means they will participate in creating in normal session creation. sunray1 is "offline", which means it will not participate in session creation during load balancing for this failover group, but sessions can still be created on it explicitly using utswitch or utselect -R, or implicitly if all other servers are down. The LAN interfaces for all of the hosts are Up, and all but sunray-sras are Available to Sun Rays (sunray-sras did not have "utadm -a" run to configure its interface for Sun Ray service, and it does not have allowLANConnections=true set in auth.props. It is a dedicated SRAS server for the failover group). Both sunray2 and sunray3 are Managing Sun Rays, because "utadm -a" was run for their LAN interfaces. They will offer DHCP parameters and possibly addresses during the Sun Ray bootup phase for Sun Rays on their local subnets.

In the second example above, all of the hosts are Trusted and oNline, meaning that they will all participate in failover and load balancing for their Available interfaces. 193.25.0.0/24 is the LAN network and the other networks are Sun Ray interconnects. All of the LAN interfaces are Up, meaning that they are reachable, but none are Available for Sun Ray service and none are Managing Sun Rays on the local subnet. All of the interconnect interfaces are Up and Available, and all are Managing Sun Rays.



**SEE ALSO** | utadm, auth.props, utswtich, utselect

**ATTRIBUTES** | See attributes(5) for descriptions of the following attributes:

Attribute Type	Attribute Value
Availability	SUNWuta



<b>NAME</b>	utinstall - Sun Ray server software installation, upgrade, and removal utility.
<b>SYNOPSIS</b>	<pre>/cdrom/cdrom0/utinstall [-a <i>admin-file</i>] [-d <i>media-dir</i>] [-u] [-q]</pre> <pre>/opt/SUNWut/sbin/utinstall [-a <i>admin-file</i>] [-d <i>media-dir</i>] [-u] [-q]</pre>
<b>DESCRIPTION</b>	<p>The <code>utinstall</code> 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.</p> <p>The <code>utinstall</code> command is run under superuser privileges and prompts the user before taking any action. Using the default administration file and media directory is recommended.</p> <p>The <code>utinstall</code> command cannot be run when the current working directory is <code>/opt/SUNWut</code> or <code>/var/opt/SUNWut</code> or <code>/etc/opt/SUNWut</code>.</p>
<b>OPTIONS</b>	<p>The following options are supported.</p> <p><code>-a <i>admin-file</i></code> Instead of the default, use the <i>admin-file</i> as an installation administration file for <code>pkgadd</code> operations (see the <code>-a</code> option for <code>pkgadd(1M)</code>). The <code>admin_default</code> file, located at the installation media root, is used by default.</p> <p><code>-d <i>media-dir</i></code> Instead of the default, use <i>media-dir</i> as the installation media root directory.</p> <p><code>-q</code> Quick install/remove. By appending this option, the <code>utinstall</code> command will automatically install or remove the software with no user interaction.</p> <p><code>-u</code> Remove previously installed Sun Ray server software.</p> <p>Without arguments, an interactive install or upgrade of the Sun Ray server software is performed.</p>
<b>FILES</b>	<p>The following files are used:</p> <ul style="list-style-type: none"> <li>■ <code>/cdrom/cdrom0/admin_default</code></li> <li>■ <code>/opt/SUNWut/etc/admin_default</code></li> </ul> <p>These are the default installation administration files used by <code>pkgadd</code> operations.</p>
<b>ATTRIBUTES</b>	See <code>attributes(5)</code> for descriptions of the following attributes:

Attribute Type	Attribute Value
Availability	SUNWuto

**SEE ALSO** | patchadd(1M), patchrm(1M), pkgadd(1M), pkgrm(1M), admin(4)

<b>NAME</b>	utkiosk - Sun Ray script to update kiosk configuration locally and within a failover group.
<b>SYNOPSIS</b>	<code>/opt/SUNWut/sbin/utkiosk {-e kiosk -i kiosk}</code>
<b>DESCRIPTION</b>	The <code>utkiosk</code> script is used to import and export kiosk configuration information into the LDAP database. <code>utkiosk</code> also updates local configuration files in the <code>/var/opt/SUNWut/kiosk</code> directory and application working files in the <code>/var/opt/SUNWut/kiosk/config</code> directory. <code>utkiosk</code> is used primarily in failover groups.
<b>OPTIONS</b>	<p>The following options are supported.</p> <p><code>-e kiosk</code>      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.</p> <p><code>-i kiosk</code>      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.</p>
<b>EXAMPLES</b>	<p><b>EXAMPLE 1</b> This command imports the configuration to LDAP and updates all local files.</p> <pre># utkiosk -i kiosk</pre> <p><b>EXAMPLE 2</b> This command exports the configuration from LDAP and updates all local files</p> <pre># utkiosk -e kiosk</pre>
<b>FILES</b>	<p>The following files are used:</p> <p><code>/var/opt/SUNWut/kiosk/kiosk.conf</code></p> <p>All application working files in <code>/var/opt/SUNWut/kiosk/config</code></p>
<b>EXIT STATUS</b>	<p>The following exit values are returned:</p> <p>0                  Success</p> <p>1                  Failure</p>

**ATTRIBUTES**

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

Attribute Type	Attribute Value
Availability	SUNWuta

<b>NAME</b>	utmhadm - Sun Ray appliance multihead group configuration utility.
<b>SYNOPSIS</b>	<pre> /opt/SUNWut/sbin/utmhadm [groupname]  /opt/SUNWut/sbin/utmhadm -a groupname -g COLSxROWS -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 </pre>
<b>DESCRIPTION</b>	<p>The <code>utmhadm</code> command provides a way to administer Sun Ray server multihead multihead groups. The information that <code>utmhadm</code> displays and that is editable is stored in the Sun Ray administration database.</p> <p>The <code>utmhadm</code> operations that only display information may be run by any user. Operations that change data must be run as superuser.</p>
<b>OPTIONS</b>	<p>The following options are supported.</p> <ul style="list-style-type: none"> <li>-a <i>groupname</i> Creates a new multihead group having identifier <i>groupname</i>. This name must be unique and not already exist on the system.</li> <li>-d <i>groupname</i> Removes the multihead group for the specified <i>groupname</i>.</li> <li>-e Populates the system multihead group database with input data of the format produced by <code>-o</code>, from standard input.</li> <li>-f <i>filename</i> Specify a <i>filename</i> for use with <code>-e</code> or <code>-o</code> instead of standard input or output.</li> <li>-g <i>COLSxROWS</i> Specifies the geometry of the multihead group in the form <i>COLSxROWS</i>. This number of columns and rows must not exceed the maximum number allowed and must match the number of appliances specified with <code>-l</code>. This option can only be used with <code>-a</code>.</li> <li>-h Prints the usage message.</li> <li>-l <i>CID1, CID2, . . . , CIDn</i> Specifies the appliance canonical identifiers when creating a group. A canonical identifier has the form IEEE802.<i>nnnnnnnnnnnn</i> or <i>nnnnnnnnnnnn</i> (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.</li> </ul>

- 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 Administrator's Guide*



**NAME** utmhconfig - Sun Ray multihead 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 [-l]

**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** utmhscreen accepts the following option:

-l Indicates to utmhscreen that it is being auto launched by the windowing system. The use of this option is beyond the scope of this manual.

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

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

**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)



<b>NAME</b>	utparallel, utserial - Sun Ray serial and parallel port device driver emulators.
<b>SYNOPSIS</b>	#include <sys/types.h> #include <fcntl.h>
<b>utserial</b>	#include <sys/termios.h> #include <termio.h>
<b>utparallel</b>	#include <sys/ecppio.h>
<b>DESCRIPTION</b>	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.
<b>EXTENDED DESCRIPTION</b>	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.
<b>API</b>	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.
<b>FILES</b>	The following files are used: <ul style="list-style-type: none"> <li>■ /dev/utserial Master port for utserial</li> <li>■ /dev/utparallel Master port for utparallel</li> </ul>
<b>ATTRIBUTES</b>	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)

<b>NAME</b>	utparalleld - Sun Ray printer service daemon.
<b>SYNOPSIS</b>	<code>/opt/SUNWut/lib/utparalleld [-D <i>debug-level</i>] [-o <i>optroot</i>] [-r]</code>
<b>DESCRIPTION</b>	<p>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.</p> <p>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.</p> <p>When a parallel adaptor or a USB printer is attached to an appliance, utparalleld creates device nodes in the /tmp/SUNWut/units/<i>CID</i> directory. <i>CID</i> is the canonical identifier for an appliance (IEEE802.<i>nnnnnnnnnnnn</i>). UTDEVROOT points to this directory through a symbolic link. A user can use \$UTDEVROOT/dev/printers/<i>printer-name</i> to access a specific printer.</p> <p>Error messages from utparalleld are logged using syslog(3), with a facility value of LOG_DAEMON.</p>
<b>OPTIONS</b>	<p>The following options are supported.</p> <p><code>-D <i>debug-level</i></code> Debug mode. Use is beyond the scope of this document.</p> <p><code>-o <i>optroot</i></code> Use <i>optroot</i> as the parallel service's root directory for device node creation. The default value is /tmp/SUNWut. <i>optroot</i> should be the same directory as the <i>optroot</i> directory used by utdevmgrp(1M).</p> <p><code>-r</code> 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.</p>
<b>FILES</b>	<p>The following files are used:</p> <p>/tmp/SUNWut                   The customary location for temporary files used by the Sun Ray server managers, designated by <i>optroot</i>.</p> <p>/tmp/SUNWut/.utdevmgr        The named pipe used for communication between the device manager and device driver services.</p>

`/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.nnnnnnnnnnnn` or `nnnnnnnnnnnn` (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)`



<b>NAME</b>	utpolicy - Sun Ray authentication manager policy management command.
<b>SYNOPSIS</b>	<pre> /opt/SUNWut/sbin/utpolicy -a [-g] [-k type] [-m] [-M] [-d] [-p] [-r type] [-s type] [-z type] [-S smartcard_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 </pre>
<b>DESCRIPTION</b>	The utpolicy command simplifies and writes the policy configuration of the Sun Ray authentication manager, utauthd(1M).
<b>OPTIONS</b>	<p>The following options are supported.</p> <p>Three categories of options are supported: Policy Setting, Card Reader Assignment, and Software Restart</p> <p>-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.</p> <p><b>POLICY SETTING</b></p> <p>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</p> <p>-g                    Turn on session selection within a server group. Allows the user to select on which server the user's session is run.</p> <p>-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.</p>

- 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.
  - d Disable the `Exit` menu option from the Non-smartcard mobility GUI. The `Exit` option is enabled by default.
  - 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.
  - S *smartcard\_type* Specify the smartcard type that should cause the Smartcard Login GUI (`utsclogin`) to be invoked when a card of the type in the list is inserted into the DTU. This is the smartcard token type (without the token ID) as provided by the Sun Ray software. If more than one smartcard type is to be specified, multiple -S options must be specified.
- Note** – in order for the Smartcard Login GUI to operate properly, the `/etc/pam.conf` file must have at least one PAM module that supports PAM authentication configured for the `utsclogin` PAM client and using the PAM client name "`utsclogin-<smartcard_type>`". `utpolicy` does not configure nor check for proper configuration of the `/etc/pam.conf` file.

#### CARD READER ASSIGNMENT

**Note** – Card Reader Assignment through the `-t` option to `utpolicy` has been deprecated in the 2.0 release. The new `utreader` command is the appropriate tool for performing these operations.

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

**Note** – The Software Restart function through the `-i` to `utpolicy` has been deprecated in 2.0. The new `utrestart` is the appropriate tool for performing these operations.

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
```

**EXAMPLE 5** This command allows all appliances to be used with or without a smart card. Non-smart card mobile sessions are enabled. The smart card types "JavaBadge" and "Payflex" cause the Smartcard Login GUI (`utsclogin`) to be invoked when a card of these types is inserted into the DTU.

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

**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)`, `utreader(1M)`, `utrestart(1M)`, `auth.props(4)`, `pam(3PAM)`



<b>NAME</b>	utpreserve - Sun Ray configuration file preservation utility.
<b>SYNOPSIS</b>	<code>/cdrom/cdrom0/utpreserve [-d <i>preserve-directory</i>]</code>
<b>DESCRIPTION</b>	The <code>utpreserve</code> command stops Sun Ray services, terminates user sessions, and saves existing Sun Ray server configuration data into a compressed tar file, <code>/var/tmp/SUNWut.upgrade/preserve_1.3.tar.Z</code> .
<b>OPTIONS</b>	The following option is supported.  <code>-d <i>preserve-directory</i></code> Save the compressed tar file into the <i>preserve-directory</i> .
<b>SEE ALSO</b>	<code>utinstall(1M)</code> , <code>utconfig(1M)</code>





<b>NAME</b>	utpw - Sun Ray administration password change utility.				
<b>SYNOPSIS</b>	<b>/opt/SUNWut/sbin/utpw</b>				
<b>DESCRIPTION</b>	<p>The <code>utpw</code> 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.</p> <p><code>utpw</code> changes the password both in the administration database, and the password file on the local server.</p> <p>In a failover group, <code>utpw</code> 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 <code>utpw</code> on them to change the password files.</p>				
<b>OPTIONS</b>	There are no options for <code>utpw</code> .				
<b>EXAMPLES</b>	<p><b>EXAMPLE 1</b> This command changes the administration password:</p> <pre># /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.</pre>				
<b>FILES</b>	<p>The following files are used:</p> <ul style="list-style-type: none"> <li>■ <code>/etc/opt/SUNWut/utadmin.pw</code></li> <li>■ <code>/etc/opt/SUNWut/utadmin.conf</code></li> </ul>				
<b>EXIT STATUS</b>	<p>The following exit values are returned:</p> <table border="0"> <tr> <td style="padding-right: 20px;">0</td> <td>Success</td> </tr> <tr> <td>1</td> <td>Error</td> </tr> </table>	0	Success	1	Error
0	Success				
1	Error				
<b>ATTRIBUTES</b>	<p>See <code>attributes(5)</code> for descriptions of the following attributes:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Attribute Type</th> <th style="text-align: center;">Attribute Value</th> </tr> </thead> <tbody> <tr> <td>Availability</td> <td>SUNWuta</td> </tr> </tbody> </table>	Attribute Type	Attribute Value	Availability	SUNWuta
Attribute Type	Attribute Value				
Availability	SUNWuta				
<b>SEE ALSO</b>	<code>utdesktop(1M)</code> , <code>utuser(1M)</code> , <i>Sun Ray Server Software Administrator's Guide</i>				

**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.

<b>NAME</b>	utquery - query Sun Ray Desktop Units current parameter values
<b>SYNOPSIS</b>	<pre>/opt/SUNWut/sbin/utquery [-d] IP_address /opt/SUNWut/sbin/utquery -h</pre>
<b>DESCRIPTION</b>	<p>The <code>utquery</code> command allows administrators to query the current parameter values on a Sun Ray Desktop Unit (DTU). The <code>IP_address</code> may specify the network IP address of a single DTU to query, a subnet broadcast address to query all DTUs on the subnet or the broadcast address to query all DTUs associated with this Sun Ray server.</p> <p>The <code>utquery</code> command can be used to aid in diagnosing problems when a DTU is unable to successfully connect to an authentication manager or when a Multihead group is redirected outside their "home" failover group.</p>
<b>OPTIONS</b>	<p>The following options are supported.</p> <p><code>-d</code> Report the DHCP parameters the DTU obtained at boot-up.</p> <p><code>-h</code> Print the usage.</p>
<b>EXAMPLES</b>	<p><b>EXAMPLE 1</b> To display the DHCP values for the DTU at IP address 129.146.58.182</p> <pre>% utquery -d 129.146.58.182</pre> <p>The following is an example of the output for the command above:</p> <pre>terminalID=080020d87c95 terminalIPA=129.146.58.182 Subnet=255.255.255.0 Router=129.146.58.1 MTU=1500 Broadcast=129.146.58.255 LeaseTim=3600 DHCPServer=129.146.58.29 INFORMServer=129.146.58.136 AuthSrvr=129.146.58.136 AuthPort=7009 LogHost=129.146.58.136 FwSrvr=129.146.58.95 NewTVer=2.0_03.b,REV=2002.04.18.16.26 currentAuth=129.146.58.136 currentFW=2.0_19.c,REV=2002.09.06.15.54</pre> <p><b>EXAMPLE 2</b> To display the DHCP values for all DTUs on subnet 129.146.58.:</p> <pre>utquery -d 129.146.58.255</pre>

**EXAMPLE 3** To display the DHCP values for all DTUs on this server:

```
utquery -d 255.255.255.255
```

## ATTRIBUTES

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

Attribute Type	Attribute Value
Availability	SUNWuta

## NOTES

The `utquery` command will only report on DTUs that are operating with firmware version 2.0 or newer.

The `utquery` command will only report on DTUs that have successfully acquired DHCP parameters from a DHCP server.

The `NewTVer` value reported is the firmware revision level supplied by DHCP, which is used to determine whether a firmware upgrade is required. The `currentFW` value is the firmware revision level of the current firmware running in the Sun Ray device.

The `AuthSrvr` and `AltAuth` values reported are those provided by DHCP parameters at appliance boot time. The `currentAuth` value is the IP address of the Sun Ray server to which the Sun Ray is currently connected.

<b>NAME</b>	<code>utrcmd</code> - Sun Ray remote administration utility.
<b>SYNOPSIS</b>	<code>/opt/SUNWut/lib/utrcmd [-n] hostname command [args]</code>
<b>DESCRIPTION</b>	<p>The <code>utrcmd</code> program provides a way to run Sun Ray administration commands remotely. The <code>utrcmd</code> program contacts the <code>in.utrcmdd</code> daemon on the remote <i>hostname</i> and executes the specified <i>command</i> with the specified arguments <i>args</i> (if any).</p> <p><code>utrcmd</code> copies its standard input to the specified command, the standard output of the command to <code>utrcmd</code>'s standard output, and the standard error of the command to <code>utrcmd</code>'s standard error. Interrupt, quit, and terminate signals are propagated to the specified command; <code>utrcmd</code> terminates normally when the command does.</p>
<b>OPTIONS</b>	<p>The following option is supported.</p> <p><code>-n</code>                 Redirect the input of <code>utrcmd</code> to <code>/dev/null</code>. This option prevents interactions between <code>utrcmd</code> and the shell which invokes it. For example, if you are running <code>utrcmd</code> and invoke a <code>utrcmd</code> 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 <code>-n</code> option prevents this behavior.</p>
<b>USAGE</b>	<p>Official hostnames or nicknames may be given as the <i>hostname</i>.</p> <p>The <code>utrcmd</code> and <code>in.utrcmdd</code> programs use the Sun Ray failover group configuration to perform a set of checks before allowing the specified command to proceed.</p> <p>The program <code>utrcmd</code> runs with set-user-ID permission of root or superuser. The <code>utrcmd</code> command will only proceed if all of the following are true (on the initiating system):</p> <ul style="list-style-type: none"> <li>■ The user's real user-ID is superuser, or the user has membership rights in the <code>utadmin</code> group.</li> <li>■ The <code>auth.props</code> file is owned by superuser and is not writable by anyone other than superuser.</li> <li>■ The <code>gmSignatureFile</code> property of <code>auth.props</code> specifies a group signature file.</li> <li>■ The group signature file exists and is owned by superuser and is not readable, writable, or executable by anyone other than superuser.</li> <li>■ The group signature file is at least 8 bytes long and has similar content diversity characteristics as required by <code>passwd(1)</code>.</li> <li>■ The "<code>utrcmd/tcp</code>" service is enabled.</li> </ul>

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 its 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.





<b>NAME</b>	utreader - Sun Ray utility for configuring a terminal as a token reader.												
<b>SYNOPSIS</b>	<pre> /opt/SUNWut/sbin/utreader /opt/SUNWut/sbin/utreader-a /opt/SUNWut/sbin/utreader-c /opt/SUNWut/sbin/utreader-d /opt/SUNWut/sbin/utreader-h </pre>												
<b>DESCRIPTION</b>	<p>The <code>utreader</code> command is used to configure Sun Ray terminals as token card readers.</p> <p>When <code>utreader</code> is run with no options, the list of terminals configured as token readers is displayed. The token reader list can be viewed by all users, but any changes made to the configuration must be made by the super-user.</p> <p><b>Note</b> – Whenever a change to a the token reader configuration been made, a restart of services has to be initiated in order for the new configuration to take effect.</p>												
<b>OPTIONS</b>	<p>The following options are supported.</p> <table border="0"> <tr> <td style="padding-right: 20px;">-a</td> <td>Configures the terminal with the specified terminal ID as a token reader.</td> </tr> <tr> <td style="padding-right: 20px;">&lt;terminalId&gt;</td> <td></td> </tr> <tr> <td style="padding-right: 20px;">-c</td> <td>Unconfigures all token readers and restores them as normal Sun Ray terminals.</td> </tr> <tr> <td style="padding-right: 20px;">-d</td> <td>Unconfigures the token reader with the specified terminal ID and restores it as a normal Sun Ray terminal.</td> </tr> <tr> <td style="padding-right: 20px;">&lt;terminalID&gt;</td> <td></td> </tr> <tr> <td style="padding-right: 20px;">-h</td> <td>Displays the usage message.</td> </tr> </table>	-a	Configures the terminal with the specified terminal ID as a token reader.	<terminalId>		-c	Unconfigures all token readers and restores them as normal Sun Ray terminals.	-d	Unconfigures the token reader with the specified terminal ID and restores it as a normal Sun Ray terminal.	<terminalID>		-h	Displays the usage message.
-a	Configures the terminal with the specified terminal ID as a token reader.												
<terminalId>													
-c	Unconfigures all token readers and restores them as normal Sun Ray terminals.												
-d	Unconfigures the token reader with the specified terminal ID and restores it as a normal Sun Ray terminal.												
<terminalID>													
-h	Displays the usage message.												
<b>EXAMPLES</b>	<p><b>EXAMPLE 1</b> This command configures a terminal as a token reader where the terminal ID is AAAABBBBCCCC.</p> <pre>% utreader -a AAAABBBBCCCC</pre> <p><b>EXAMPLE 2</b> This command unconfigures all token readers.</p> <pre>utreader -c</pre>												
<b>EXIT STATUS</b>	<p>The following exit values are returned:</p> <table border="0"> <tr> <td style="padding-right: 20px;">0</td> <td>Success</td> </tr> <tr> <td style="padding-right: 20px;">1</td> <td>Failure</td> </tr> </table>	0	Success	1	Failure								
0	Success												
1	Failure												

**ATTRIBUTES**

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

Attribute Type	Attribute Value
Availability	SUNWuta

<b>NAME</b>	utreplica - LDAP replication utility for Sun Ray servers.
<b>SYNOPSIS</b>	<pre> /opt/SUNWut/sbin/utreplica -p [-a   -d]secondary-server1 [secondary-server2]... /opt/SUNWut/sbin/utreplica -s primary-server /opt/SUNWut/sbin/utreplica -l /opt/SUNWut/sbin/utreplica -u /opt/SUNWut/sbin/utreplica -z [port#] </pre>
<b>DESCRIPTION</b>	The <code>utreplica</code> 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.
<b>OPTIONS</b>	<p>The following options are supported.</p> <ul style="list-style-type: none"> <li>-l List the current failover administration status.</li> <li>-p [-a -d] <i>secondary-server</i> Configure the primary server. <i>secondary-server</i> is the host name of the secondary server. List all secondary servers within the failover group. <ul style="list-style-type: none"> <li>-a is used to add the specified secondary servers to the current list of secondary servers.</li> <li>-dis used to delete the specified secondary servers from the current list of secondary servers.</li> </ul> </li> <li>-s <i>primary-server</i> Configure a secondary server. <i>primary-server</i> is the host name of the primary server.</li> <li>-u Unconfigure this Sun Ray server for LDAP database replication.</li> <li>-z [<i>port#</i>] update the port number with the specified one for the datastore service. If it's run on the primary server without specifying the port number, it will just update all the necessary configuration files on the primary with the default port for the datastore service. If it's run on the secondary server without specifying the port number, it will resync all necessary configuration files on the secondary server with the port number that's currently configured on the primary server.</li> </ul>
<b>USAGE</b>	<code>utreplica</code> is used only on Sun Ray servers in a failover group. Configure the primary Sun Ray server first, then the secondary servers.
<b>FILES</b>	The following files are configured on the primary Sun Ray server:

- /etc/opt/SUNWut/srds/current/utdsd.conf
- /etc/opt/SUNWut/srds/current/utdsd.ini

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

- /etc/opt/SUNWut/srds/current/utdsd.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.

<b>NAME</b>	utresadm - manage explicit monitor resolution settings for Sun Ray.
<b>SYNOPSIS</b>	<pre> /opt/SUNWut/sbin/utresadm /opt/SUNWut/sbin/utresadm -a -c CID -t token resname /opt/SUNWut/sbin/utresadm -d -c CID -t token /opt/SUNWut/sbin/utresadm -p [-c CID ] [-t token] /opt/SUNWut/sbin/utresadm -o /opt/SUNWut/sbin/utresadm -i </pre>
<b>DESCRIPTION</b>	<p>The <code>utresadm</code> command allows an administrator to create, delete and view an explicit resolution for a monitor attached to a Sun Ray information appliance. Resolutions established through <code>utresadm</code> take precedence over any resolution discovered through a DDC exchange between the Sun Ray unit and the monitor. Resolutions may be specified for a particular combination of Sun Ray and access token, for a particular Sun Ray independent of access token, and for all Sun Rays controlled by servers in a failover group. If multiple configuration records could apply to a given session then the most specific match is applied.</p> <p><code>utresadm</code> with no options or arguments shows the resolution that has been configured for the current access token in the current Sun Ray unit.</p> <p>Available resolution names are reported by <code>utresdef(1M)</code>.</p>
<b>OPTIONS</b>	<p>The following options are supported.</p> <ul style="list-style-type: none"> <li>-a establishes the resolution resname as the preferred resolution for the given CID and token. If token is specified as default then resname becomes the preferred resolution for all tokens at the given CID for which no explicit resolution has been configured. If both CID and token are specified as default then resname becomes the preferred resolution for all CIDs and tokens for which no explicit resolution has been configured. This option is available only to the superuser.</li> <li>-d deletes an explicit resolution previously configured for the given CID and token. This option is available only to the superuser.</li> <li>-p shows the explicit resolution configured for the given CID and token. If no token is specified then all configuration records for the given CID are shown. If no CID is specified then all configuration records for the given token are shown. If neither CID nor token is specified then all configuration records are shown.</li> <li>-o shows all known resolution configuration records in a format suitable for consumption by <code>utresadm -i</code>.</li> </ul>

- i reads (from standard input) a list of resolution configuration records in the form emitted by utresname -o and configures the corresponding resolutions. This option is available only to the superuser.
- c the canonical ID of the Sun Ray desktop unit, or the word default if this operation is to apply to all desktop units.
- t the access token for the session, or the word default if this operation is to apply to all tokens.

**OPERANDS**

*resname* The desired resolution for the given token at the given CID.

**EXIT STATUS**

The following exit values are returned:

- 0 if the desired activity was completed without error.
- 1 if the command terminated because of a command line syntax problem.
- 2 if the Sun Ray data store was inaccessible.
- 3 if the resolution definition (provided on standard input) was unacceptable.

**FILES**

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

**ATTRIBUTES**

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

Attribute Type	Attribute Value
Availability	SUNWuta

**SEE ALSO**

*utsettings(1)*, *utset(1)*, *utresdef(1M)*, *utconfig(1M)*

**NOTES**

utresadm provides no assurance that the configured resolution timing can be understood by the monitor attached to a Sun Ray unit; it is possible to

configure a resolution that can not be understood by the monitor. *utsettings(1)* provides a safer interactive means of establishing an explicit resolution for the current access token on the current Sun Ray unit.

utresadm is available only after *utconfig(1M)* has been run to activate the Sun Ray data store.

<b>NAME</b>	utresdef - manage monitor resolution definitions for Sun Ray.
<b>SYNOPSIS</b>	<pre> /opt/SUNWut/sbin/utresdef /opt/SUNWut/sbin/utresdef <i>resname</i> /opt/SUNWut/sbin/utresdef -a [-c <i>comment</i>] <i>dimensions resname</i> /opt/SUNWut/sbin/utresdef -d <i>resname</i> /opt/SUNWut/sbin/utresdef -o /opt/SUNWut/sbin/utresdef -i /opt/SUNWut/sbin/utresdef -h </pre>
<b>DESCRIPTION</b>	<p>The <code>utresdef</code> command allows an administrator to create, delete and view resolution definitions for monitors attached to Sun Ray information appliances.</p> <p>Resolutions are associated with specific Sun Ray units by <code>utresadm(1M)</code>. Users may configure resolutions for their own personal access tokens through <code>utsettings(1)</code> or <code>utset(1)</code>.</p> <p>Available resolution names and their definitions are reported by <code>utresdef(1M)</code> when nothing but the command is entered. If a <i>resname</i> is specified without any options, its definition is reported.</p>
<b>OPTIONS</b>	<p>The following options are supported.</p> <ul style="list-style-type: none"> <li>-a defines the resolution <i>resname</i>. This option is available only to the superuser.</li> <li>-d deletes the resolution definition named <i>resname</i>. This option is available only to the superuser.</li> <li>-o shows all known resolution definitions in a format suitable for consumption by <code>utresdef -i</code>.</li> <li>-i reads (from standard input) a list of <i>resnames</i>, <i>comments</i>, <i>dimensions</i> and resolution definitions in the form emitted by <code>utresdef -o</code> and configures the corresponding resolutions. This option is available only to the superuser.</li> <li>-c a comment explaining the nature and purpose of the resolution being defined.</li> <li>-h Displays the usage message for this command.</li> </ul>

**OPERANDS**

- dimensions* the dimensions, in the form widthxheight, of the on-screen display after this resolution definition has been applied to the Sun Ray desktop unit. The dimensions are often implied by *resname* but this is not guaranteed. *resname* may take some unconventional form (it might perhaps be named for a particular model of monitor) from which the dimensions can not be inferred, so the resulting dimensions must be provided explicitly.
- resname* the name of the resolution being defined, deleted or displayed. *resname* is conventionally given in the form widthxheight@rate (e.g. 640x480@60) but this convention is not enforced.

**EXIT STATUS**

The following exit values are returned:

- 0 if the desired activity was completed without error.
- 1 if the command terminated because of a command line syntax problem.
- 2 if the Sun Ray data store was inaccessible.
- 3 if the resolution definition (provided on standard input) was unacceptable.

**FILES**

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

**ATTRIBUTES**

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

Attribute Type	Attribute Value
Availability	SUNWuta

**SEE ALSO**

`utsettings(1)`, `utset(1)`, `utresadm(1M)`, `utconfig(1M)`

**NOTES**

`utresdef` is available only after `utconfig(1M)` has been run to activate the Sun Ray data store.



**NAME**        `utrestart` - Sun Ray utility for resetting and restarting services.

**SYNOPSIS**    `/opt/SUNWut/sbin/utrestart`  
`/opt/SUNWut/sbin/utrestart -c`  
`/opt/SUNWut/sbin/utrestart -h`

**DESCRIPTION**    The `utrestart` command is used for resetting and restarting Sun Ray services. It replaces the `utpolicy -i` option which has been deprecated in 2.0.  
`utrestart` can only be run by the super-user.

**OPTIONS**        The following options are supported.

`-c`        Restarts Sun Ray services. Sessions will be lost.

`-h`        prints the usage for this command.

**EXAMPLES**        **EXAMPLE 1**    This resets services.  
                   `# /opt/SUNWut/sbin/utrestart`

**EXAMPLE 2**    This restarts services.  
                   `# /opt/SUNWut/sbin/utrestart -c`

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

Attribute Type	Attribute Value
Availability	SUNWuta



<b>NAME</b>	<code>utselect</code> - Sun Ray failover group server selection tool.				
<b>SYNOPSIS</b>	<code>/opt/SUNWut/bin/utselect [-L] [-R] [-S] [-X]</code>				
<b>DESCRIPTION</b>	The <code>utselect</code> command is a graphical user interface (GUI) to the <code>utswitch</code> 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 <code>utswitch -l</code> command and updates the information displayed in the GUI. The Ok button executes a <code>utswitch -h</code> command to the server highlighted.				
<b>OPTIONS</b>	<p>The following options are supported.</p> <p><code>-L</code> Configures <code>utselect</code> to run in “login” mode before the CDE log in screen is displayed. Where:</p> <ul style="list-style-type: none"> <li>■ If only one server is available, the command exits.</li> <li>■ The current server is set as the default.</li> <li>■ Selecting the current server causes the command to exit</li> <li>■ The locale is determined in a manner similar to CDE</li> <li>■ The screen is centered in the display</li> </ul> <p><code>-R</code> Remote server selection is enabled. This enables an entry field where a networked server name can be entered.</p> <p><code>-S</code> Remote server selection is disabled.</p> <p><code>-X</code> Exit after making a selection from the list.</p>				
<b>EXAMPLES</b>	<p><b>EXAMPLE 1</b> This command enables users to select which Sun Ray server or session to connect. The GUI will exit after selection</p> <pre>% /opt/SUNWut/bin/utselect -X</pre>				
<b>FILES</b>	<p>Two properties in the <code>auth.props(4)</code> file impact the operation of <code>utselect</code>. They are:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px;"><code>selectAtLogin=true</code></td> <td style="padding: 5px;">This setting will cause <code>utselect</code> to run before <code>dtlogin</code> and allow users to start their session on a particular machine. The default value is “false”</td> </tr> <tr> <td style="padding: 5px;"><code>remoteSelect=true</code></td> <td style="padding: 5px;">This setting will cause <code>utselect</code> to behave as if the <code>-R</code> option were specified. This will allow users to input the name of a server outside of the default HA group, if the Sun Ray can connect to it (i.e. the Sun Ray can be routed to the server, such as in a LAN deployment). The default value is false.</td> </tr> </table>	<code>selectAtLogin=true</code>	This setting will cause <code>utselect</code> to run before <code>dtlogin</code> and allow users to start their session on a particular machine. The default value is “false”	<code>remoteSelect=true</code>	This setting will cause <code>utselect</code> to behave as if the <code>-R</code> option were specified. This will allow users to input the name of a server outside of the default HA group, if the Sun Ray can connect to it (i.e. the Sun Ray can be routed to the server, such as in a LAN deployment). The default value is false.
<code>selectAtLogin=true</code>	This setting will cause <code>utselect</code> to run before <code>dtlogin</code> and allow users to start their session on a particular machine. The default value is “false”				
<code>remoteSelect=true</code>	This setting will cause <code>utselect</code> to behave as if the <code>-R</code> option were specified. This will allow users to input the name of a server outside of the default HA group, if the Sun Ray can connect to it (i.e. the Sun Ray can be routed to the server, such as in a LAN deployment). The default value is false.				

**ATTRIBUTES**

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

Attribute Type	Attribute Value
Availability	SUNWuto

**SEE ALSO**

`utswitch(1)`, `attributes(5)`, `auth.props(4)`

<b>NAME</b>	utserial, utparallel - Sun Ray serial and parallel port device driver emulators.
<b>SYNOPSIS</b>	#include <sys/types.h> #include <fcntl.h>
<b>utserial</b>	#include <sys/termios.h> #include <termio.h>
<b>utparallel</b>	#include <sys/ecppio.h>
<b>DESCRIPTION</b>	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.
<b>EXTENDED DESCRIPTION</b>	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.
<b>API</b>	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.
<b>FILES</b>	The following files are used: <ul style="list-style-type: none"> <li>■ /dev/utserial Master port for utserial</li> <li>■ /dev/utparallel Master port for utparallel</li> </ul>
<b>ATTRIBUTES</b>	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)

<b>NAME</b>	utseriald - Sun Ray appliance serial services daemon.
<b>SYNOPSIS</b>	<code>/opt/SUNWut/bin/utseriald [-D <i>debug-level</i>] [-o <i>optroot</i>] [-r]</code>
<b>DESCRIPTION</b>	<p>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:</p> <p><a href="http://www.sun.com/sunray">http://www.sun.com/sunray</a></p> <p>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.</p> <p>When an application opens a device node, the ldterm(7M) and ttcompat(7M) modules are pushed below the STREAM head.</p> <p>When a serial adaptor is attached to an appliance, utseriald creates device nodes in the /tmp/SUNWut/units/<i>CID</i> directory. <i>CID</i> is the canonical identifier for an appliance (IEEE802.nnnnnnnnnnnn). UTDEVROOT points to this directory through a symbolic link. A user can use \$UTDEVROOT/dev/term/<i>terminal-name</i> to access the serial ports on the Sun Ray appliance.</p> <p>Error messages from utseriald are logged using syslog(3), with a facility value of LOG_DAEMON.</p>
<b>OPTIONS</b>	<p>The following options are supported.</p> <p><code>-D <i>debug-level</i></code> Debug mode. Use is beyond the scope of this document.</p> <p><code>-o <i>optroot</i></code> Use <i>optroot</i> as the serial service's root directory for device node creation. The default value is /tmp/SUNWut. <i>optroot</i> should be the same directory as the <i>optroot</i> directory used by utdevmgrd(1M).</p> <p><code>-r</code> 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.</p>
<b>FILES</b>	<p>The following files are used:</p> <p>/tmp/SUNWut                   The customary location for temporary files used by the Sun Ray server managers, designated by <i>optroot</i>.</p> <p>/tmp/SUNWut/.utdevmgr       The named pipe used for communication between the device manager and device driver services.</p>

/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. <i>nnnnnnnnnnnn</i> or <i>nnnnnnnnnnnn</i> (the 12-digit hexadecimal MAC address of the appliance). Each directory contains a <code>dev</code> directory and a <code>devices</code> directory.
/tmp/SUNWut/units/ <i>CID</i> /dev/term	The directory containing links to serial device names for each appliance. <i>CID</i> is the canonical identifier for an appliance.
/tmp/SUNWut/sessions	The directory containing links to appliances in the <code>units</code> 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)`



<b>NAME</b>	<code>utsession</code> - List and manage the Sun Ray sessions on the local Sun Ray server.						
<b>SYNOPSIS</b>	<pre> /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 </pre>						
<b>DESCRIPTION</b>	<p>The first synopsis (<code>-p</code>) is used to print Sun Ray sessions (both active and suspended) for the specified user or token on the current server. When the <code>-u</code>, <code>-t</code>, <code>-n</code>, and <code>-d</code> options are not used, <code>utsession</code> prints all of the Sun Ray sessions on the current server.</p> <p>When listing the Sun Ray sessions, <code>utsession</code> also lists the state for each session:</p> <table border="0" style="margin-left: 20px;"> <tr> <td style="padding-right: 10px;">D</td> <td>Disconnected — The session is currently not attached to any Sun Ray. The session is considered connected if this flag is omitted.</td> </tr> <tr> <td style="padding-right: 10px;">I</td> <td>Idling — A <code>dtlogin</code> session that is currently waiting for user login (ie. <code>dtgreet</code>). User has already logged into the <code>dtlogin</code> session if the flag is omitted.</td> </tr> <tr> <td style="padding-right: 10px;">S</td> <td>Suspended — The session is currently suspended. The session is considered running if this flag is omitted.</td> </tr> </table> <p>The second synopsis (<code>-k   -s</code>) is used to kill or suspend Sun Ray sessions on the current server. At least one of the <code>-d</code>, <code>-u</code>, <code>-t</code>, or <code>-n</code> options must be specified. Unless the <code>-a</code> option is specified, more than one session matching the specified criteria will return an error.</p> <p>When suspending a session, <code>utsession</code> uses <code>dterror.ds(1)</code> to display a message on the user's screen to indicate that the session is being suspending by the administrator.</p> <p>The third synopsis (<code>-r</code>) is used to resume Sun Ray sessions on the current server. If <code>-d</code>, <code>-u</code>, <code>-t</code>, or <code>-n</code> option is not specified, <code>utsession</code> will resume all the suspended Sun Ray sessions on the current server. Unless the <code>-a</code> option is specified, more than one session matching the specified criteria will return an error.</p> <p>The fourth synopsis (<code>-h</code>) displays the usage of this command.</p> <p><b>Note</b> – This command must be run as root.</p>	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 <code>dtlogin</code> session that is currently waiting for user login (ie. <code>dtgreet</code> ). User has already logged into the <code>dtlogin</code> session if the flag is omitted.	S	Suspended — The session is currently suspended. The session is considered running if this flag is omitted.
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 <code>dtlogin</code> session that is currently waiting for user login (ie. <code>dtgreet</code> ). User has already logged into the <code>dtlogin</code> session if the flag is omitted.						
S	Suspended — The session is currently suspended. The session is considered running if this flag is omitted.						

**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)`



<b>NAME</b>	utsessiond - Sun Ray session manager daemon.
<b>SYNOPSIS</b>	<code>/opt/SUNWut/lib/utsessiond [-a <i>authlist</i>] [-c <i>authfile</i>] [-d] [-h <i>hostname</i>] [-p <i>port</i>] [-P <i>nport</i>] [-r] [-t]</code>
<b>DESCRIPTION</b>	<p>The <code>utsessiond</code> 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.</p> <p>If either the <code>-a</code> or the <code>-c</code> 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 <i>authlist</i> or <i>authfile</i> 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.</p> <p>Error messages from <code>utsessiond</code> are logged using <code>syslog(3)</code> with a facility value of <code>LOG_DAEMON</code>.</p>
<b>OPTIONS</b>	<p>The following options are supported.</p> <ul style="list-style-type: none"> <li><code>-a <i>authlist</i></code>      Add the host and port pairs specified in <i>authlist</i> to the list of permitted Authentication Managers. The format of <i>authlist</i> is a comma separated list of <i>hostname:port</i> pairs.</li> <li><code>-c <i>authfile</i></code>      Add the host and port pairs specified in the ASCII file <i>authfile</i> 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 <i>hostname</i> followed by <i>port</i> number, separated by white-space. Blank lines and any line whose first printable character is “#” are ignored.</li> <li><code>-d</code>                    Enable debugging output.</li> <li><code>-h <i>hostname</i></code>      Set the <i>hostname</i> portion of the session IDs generated by the Session Manager to the specified <i>hostname</i> 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.</li> <li><code>-p <i>port</i></code>            Set the Session Manager's listen port to the specified <i>port</i> value. The default is port 7007. This is the port by which session services and Authentication Managers contact the Session Manager.</li> <li><code>-P <i>nport</i></code>            This option is no longer used. Retained only for backward compatibility.</li> </ul>

- 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)`

<b>NAME</b>	utset - View or change the Sun Ray appliance settings.																										
<b>SYNOPSIS</b>	<code>/opt/SUNWut/bin/utset [-o arg[...]] [-i arg[...]] [-d arg[...]] [-f] [-v arg[...]]</code>																										
<b>DESCRIPTION</b>	<p>The <code>utset</code> command performs Sun Ray device settings. A Sun Ray device is capable of 5 categories of settings: Audio Output, Audio Input, Mouse, Display and Video Input. There is an option for each category. Category options are followed by an argument that specifies one or more settings to affect. The argument is a comma separated list of '<i>name=value</i>' pairs. Where <i>name</i> specifies the setting and <i>value</i> specifies the desired value.</p> <p>If <code>utset</code> is run without any options the current status of all settings is printed on stdout.</p>																										
<b>OPTIONS</b>	<p>The following options are supported.</p> <p><code>-o arg[...]</code> Set the Audio Output settings:</p> <table border="0"> <tr> <td><code>s=[a s h]</code></td> <td>Output Select: auto, speaker, headphone.</td> </tr> <tr> <td><code>v=&lt;0:31&gt;</code></td> <td>Volume: a range of 0 to 31.</td> </tr> <tr> <td><code>b=&lt;-32:32&gt;</code></td> <td>Balance: a range of -32 to +32.</td> </tr> <tr> <td><code>m=[on off]</code></td> <td>Mute: on, off (also: 1, 0 respectively).</td> </tr> <tr> <td><code>e=[on off]</code></td> <td>Stereo Enhance: on, off (also: 1, 0 respectively).</td> </tr> <tr> <td><code>T=&lt;-6:6&gt;</code></td> <td>Treble: a range of -6 to +6.</td> </tr> <tr> <td><code>B=&lt;-6:6&gt;</code></td> <td>Base: a range o -6 to +6.</td> </tr> </table> <p><code>-i arg[...]</code> Set the Audio Input settings:</p> <table border="0"> <tr> <td><code>s=[m l]</code></td> <td>Input Select: microphone, lline.</td> </tr> <tr> <td><code>g=&lt;0:75&gt;</code></td> <td>Microphone Gain: a range of 0 to 75.</td> </tr> <tr> <td><code>l=&lt;0:15&gt;</code></td> <td>Line in Gain Left: a range of 0 to 15.</td> </tr> <tr> <td><code>r=&lt;0:15&gt;</code></td> <td>Line in Gain Right: a range of 0 to 15.</td> </tr> <tr> <td><code>v=&lt;0:64&gt;</code></td> <td>Volume: a range of 0 to 64.</td> </tr> <tr> <td><code>b=&lt;-32:32&gt;</code></td> <td>Balance: a range of -32 to +32.</td> </tr> </table>	<code>s=[a s h]</code>	Output Select: auto, speaker, headphone.	<code>v=&lt;0:31&gt;</code>	Volume: a range of 0 to 31.	<code>b=&lt;-32:32&gt;</code>	Balance: a range of -32 to +32.	<code>m=[on off]</code>	Mute: on, off (also: 1, 0 respectively).	<code>e=[on off]</code>	Stereo Enhance: on, off (also: 1, 0 respectively).	<code>T=&lt;-6:6&gt;</code>	Treble: a range of -6 to +6.	<code>B=&lt;-6:6&gt;</code>	Base: a range o -6 to +6.	<code>s=[m l]</code>	Input Select: microphone, lline.	<code>g=&lt;0:75&gt;</code>	Microphone Gain: a range of 0 to 75.	<code>l=&lt;0:15&gt;</code>	Line in Gain Left: a range of 0 to 15.	<code>r=&lt;0:15&gt;</code>	Line in Gain Right: a range of 0 to 15.	<code>v=&lt;0:64&gt;</code>	Volume: a range of 0 to 64.	<code>b=&lt;-32:32&gt;</code>	Balance: a range of -32 to +32.
<code>s=[a s h]</code>	Output Select: auto, speaker, headphone.																										
<code>v=&lt;0:31&gt;</code>	Volume: a range of 0 to 31.																										
<code>b=&lt;-32:32&gt;</code>	Balance: a range of -32 to +32.																										
<code>m=[on off]</code>	Mute: on, off (also: 1, 0 respectively).																										
<code>e=[on off]</code>	Stereo Enhance: on, off (also: 1, 0 respectively).																										
<code>T=&lt;-6:6&gt;</code>	Treble: a range of -6 to +6.																										
<code>B=&lt;-6:6&gt;</code>	Base: a range o -6 to +6.																										
<code>s=[m l]</code>	Input Select: microphone, lline.																										
<code>g=&lt;0:75&gt;</code>	Microphone Gain: a range of 0 to 75.																										
<code>l=&lt;0:15&gt;</code>	Line in Gain Left: a range of 0 to 15.																										
<code>r=&lt;0:15&gt;</code>	Line in Gain Right: a range of 0 to 15.																										
<code>v=&lt;0:64&gt;</code>	Volume: a range of 0 to 64.																										
<code>b=&lt;-32:32&gt;</code>	Balance: a range of -32 to +32.																										

- `-d arg[...][-f]` Set the Display settings:
- `r=<WxH@F>` Resolution/Refresh Rate: Where *W* is the desired width, *H* is the desired height and *F* is the desired refresh rate. User will be prompted to confirm that the specified Resolution/Refresh rate is satisfactory.
  - `-f` Force the specified Resolution/Refresh Rate setting, user is not prompted.
- `-v arg[...]` Set the Video Input settings:
- `b=<0:255>` Brightness: a range of 0 to 255.
  - `c=<0:63>` Contrast: a range of 0 to 63.
  - `C=<0:127>` Color: a range of 0 to 127.
  - `t=<0:255>` Tint: a range of 0 to 255.
  - `f=<0:3>` Filter: a range of 0 to 3.
  - `T=[on|off]` Color Trap: on, off (also: 1, 0 respectively).

The following is the list of supported resolutions:

"640x480@60"  
 "640x480@85"  
 "800x600@60"  
 "800x600@85"  
 "1024x768@60"  
 "1024x768@75"  
 "1024x768@85"  
 "1152x900@66"  
 "1152x900@76"  
 "1280x1024@60"  
 "1280x1024@66"  
 "1280x1024@75"  
 "1280x1024@76"  
 "1280x1024@85"

## EXAMPLES

**EXAMPLE 1** This command displays the settings for the Sun Ray appliance you are cur-



rently logged into.

% **utset**

**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** , `utsettings(1)`, `dtlogin(1X)`, `dtsession(1X)`



<b>NAME</b>	utsettings - View or change the Sun Ray appliance settings.
<b>SYNOPSIS</b>	<code>/opt/SUNWut/bin/utsettings [-H [-k <i>hotkey</i>]]</code>
<b>DESCRIPTION</b>	<p>The <code>utsettings</code> 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.</p> <p>The <code>utsettings</code> 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 <code>utsettings</code> application to follow. With each session move, the <code>utsettings</code> application displays the current appliance's configuration.</p> <p>By default, the Sun Ray server starts an instance of <code>utslaunch(1M)</code> for each session created by the user logging in via <code>dtlogin</code>. 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.</p> <p>Users can initiate similar functionality by running <code>utsettings</code> with the <code>-H</code> flag. The hotkey can be specified using the <code>-k</code> option. Only one instance of <code>utsettings -H</code> or <code>utslaunch</code> can be running per session.</p>
<b>OPTIONS</b>	<p>The following options are supported.</p> <p><code>-H</code> Start the <code>utsettings</code> application in "hotkey" mode. The <code>utsettings</code> 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 <code>utsettings.hotkey</code> property in the files listed below. See <b>FILES</b></p> <p><code>-k <i>hotkey</i></code> Use the specified key or keys as the hotkey combinations when the <code>-H</code> option is specified. This option is dependent upon the <code>-H</code> option.</p>
<b>EXAMPLES</b>	<p><b>EXAMPLE 1</b> This command displays the settings for the Sun Ray appliance you are currently logged into.</p> <pre>% utsettings</pre> <p><b>EXAMPLE 2</b> This command displays the settings for the Sun Ray appliance you are currently logged into when you press the Shift + ESC hotkey key combination.</p> <pre>% utsettings -H -k "Shift ESC"</pre>
<b>FILES</b>	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 Type	Attribute Value
Availability	SUNWuto

**SEE ALSO**

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

<b>NAME</b>	<code>utslaunch</code> - Sun Ray appliance launch application.				
<b>SYNOPSIS</b>	<code>/opt/SUNWut/lib/utslaunch</code>				
<b>DESCRIPTION</b>	<p>The <code>utslaunch</code> application is used to launch various Sun Ray applications via a "hotkey" key combination. The applications are enabled when the key combination is pressed.</p> <p>The <code>utslaunch</code> application provides hotkey functionality while consuming fewer system resources.</p> <p>Hotkey key combinations are defined in the <code>utslaunch.properties</code> files.</p>				
<b>OPTIONS</b>	There are no options for <code>utslaunch</code> .				
<b>EXAMPLES</b>	<p><b>EXAMPLE 1</b> This command starts the <code>utslaunch</code> daemon in the background.</p> <pre># utslaunch &amp;</pre>				
<b>FILES</b>	<p>The following files are used for hotkey configuration:</p> <ul style="list-style-type: none"><li>■ <code>/etc/opt/SUNWut/utslaunch_defaults.properties</code> site-wide defaults</li><li>■ <code>~/.utslaunch.properties</code> user's defaults</li><li>■ <code>/etc/opt/SUNWut/utslaunch_mandatory.properties</code> site-wide mandatory defaults</li></ul> <p>The following file is used:</p> <pre>/usr/dt/config/Xsession.d/0100.SUNWut</pre>				
<b>ENVIRONMENT VARIABLES</b>	<p><code>utslaunch</code> uses the <code>DISPLAY</code> environment variable to get the user's X display number.</p> <p>It also uses the <code>HOME</code> environment variable to get the user's home directory to be able to use user's hotkey settings.</p>				
<b>EXIT STATUS</b>	<p>The following exit values are returned:</p> <table><tr><td>0</td><td>Success</td></tr><tr><td>1</td><td>Failure</td></tr></table>	0	Success	1	Failure
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)`, `utdetach(1)`

<b>NAME</b>	utslaunch.properties - Default hotkey key combinations for various applications supported by utslaunch.
<b>SYNOPSIS</b>	<p><b>/etc/opt/SUNWut/utslaunch_defaults.properties</b></p> <p><b>~/.utslaunch.properties</b></p> <p><b>/etc/opt/SUNWut/utslaunch_mandatory.properties</b></p>
<b>DESCRIPTION</b>	<p>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:</p> <p><i>name=value</i></p> <p>where <i>name</i> is the property name and <i>value</i> is the value set.</p>
<b>EXTENDED DESCRIPTION</b>	<p>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.</p> <ol style="list-style-type: none"> <li>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.</li> <li>2. ~/.utslaunch.properties This file contains the user's default properties. These properties override application and site-wide default properties.</li> <li>3. /etc/opt/SUNWut/utslaunch_mandatory.properties This file contains site-wide mandatory default properties which supersede any application, site-wide, or user defaults.</li> </ol>
<b>PROPERTIES</b>	<p>The supported application properties are listed below. For each property, the name, description, application default, and some examples are given.</p> <p>Name — utdetach.hotkey</p> <p>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).</p> <p>Application Default — Shift Pause (Hold down Shift and press the Pause key)</p> <p>Name — utsettings.hotkey</p>

**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)`



**NAME**        `utsunmc` - Sun Ray server software 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 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 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 module.

Without arguments, addition and load of the Sun Ray server software 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.



<b>NAME</b>	<code>utswitch</code> - Sun Ray server selection and session listing utility.
<b>SYNOPSIS</b>	<code>/opt/SUNWut/bin/utswitch</code> <code>{-l   -t   -h <i>hostname</i>}</code> <code>[-k <i>token</i>]</code> <code>[-p <i>port</i>]</code> <code>[-r]</code> <code>[-s <i>sid</i>]</code>
<b>DESCRIPTION</b>	The <code>utswitch</code> 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: <code>-l</code> , <code>-t</code> , or <code>-h</code> . The <code>utselect(1)</code> command implements a GUI-based interface to this command.
<b>OPTIONS</b>	The following options are supported.
<code>-h <i>hostname</i></code>	Force an explicit switch to the server with <i>hostname</i> .
<code>-k <i>token</i></code>	Specify the token ID <i>token</i> 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.
<code>-l</code>	List the servers accessible from the current Sun Ray appliance for the current token and show any existing sessions on those servers. <ul style="list-style-type: none"> <li>■ The first field of the output is the server name.</li> <li>■ The second field is the X display number for an active user session. If no active user session exists, then <code>-1</code> is printed or <code>-2</code> is printed if the login screen is being displayed.</li> <li>■ The third field is the last connection time to an existing session, as a time value from the <code>time(2)</code> system call. If there is no session, the third field indicates status from the host as: <ul style="list-style-type: none"> <li><code>-1</code> Server is up, but there is no session.</li> <li><code>-2</code> No response received from the server.</li> <li><code>-3</code> No path from the Sun Ray to the server.</li> </ul> </li> <li>■ The fourth field is 1 if the server is offline and 0 otherwise.</li> </ul>
<code>-p <i>port</i></code>	Sets the port number of the Authentication Manager on the Sun Ray server to <i>port</i> , instead of the default 7009.
<code>-r</code>	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 <code>-h</code> 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)`

<b>NAME</b>	<code>utuser</code> - Sun Ray user administration utility.
<b>SYNOPSIS</b>	<pre> /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 -L -g  /opt/SUNWut/sbin/utuser -o  /opt/SUNWut/sbin/utuser -p tokenID  /opt/SUNWut/sbin/utuser -r token-reader </pre>
<b>DESCRIPTION</b>	<p>The <code>utuser</code> command allows the administrator to manage users registered on the Sun Ray server which the command is run. The information that <code>utuser</code> provides is from the Sun Ray administration database and the Sun Ray Authentication Manager.</p>

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).
- L -g List all users currently logged in and the servers into which they are logged in.
- 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 /tmp/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.



<b>NAME</b>	utwall - Sun Ray user notification utility.
<b>SYNOPSIS</b>	<pre> /opt/SUNWut/sbin/utwall -a <i>aufile</i> [-r <i>n</i>] [-v]{ALL   user [:<i>display</i>]   :<i>display</i>...} /opt/SUNWut/sbin/utwall [-d] [-m "<i>subject</i>"] [-t "<i>message-text</i>"] [-v]{ALL   user [:<i>display</i>]   :<i>display</i>...}  /opt/SUNWut/sbin/utwall [-u "<i>message-text</i>"] </pre>
<b>DESCRIPTION</b>	<p>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.</p> <p>Options -a and -d require superuser privileges.</p>
<b>OPTIONS</b>	<p>The following options are supported.</p> <ul style="list-style-type: none"> <li>-a <i>aufile</i>           Annunciate mode. Plays the audio file <i>aufile</i> on the specified user's X session. Audio files of type .au can be found at /usr/demo/SOUND/sounds.</li> <li>-d                    Pop up a dterror.ds window with the supplied message on each Xsun instance.</li> <li>-m "<i>subject</i>"        Send mail with the given subject "<i>subject</i>" and supplied message. If the text has white space, use single or double quotes. Substitution is supported.</li> <li>-r <i>n</i>                Repeat the annunciation <i>n</i> times. This option can only be used with -a. Default is 1.</li> <li>-t "<i>message-text</i>" Message text. Alternatively, the message can be supplied as stdin. If the text has white space, use single or double quotes. Substitution is supported.</li> <li>-u "<i>message-text</i>" If no message-text is provided, utwall clears any message text that is in the /etc/motd file.</li> <li>-v                    Verbose mode.</li> </ul>
<b>OPERANDS</b>	<p>The following operands are supported:</p> <ul style="list-style-type: none"> <li>ALL                  Action is performed on all user having an Xsun process.</li> <li>user :<i>display</i>       Action is performed on the given users (optional display number <i>display</i>) having an Xsun process.</li> <li>:<i>display</i>            Action is performed on the users having display number <i>display</i>.</li> </ul>

**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.

<b>NAME</b>	utxconfig - Sun Ray appliance X server configuration utility.
<b>SYNOPSIS</b>	<pre> /opt/SUNWut/bin/utxconfig [-a] [-c <i>config-file</i>] [-d <i>display</i>] [-D] [-l] [-L <i>geom</i>] [-m <i>multihead</i>] [-p <i>pcolor</i>] [-r <i>res</i>] [-R <i>geom</i>] [-s <i>asize</i>] [-S <i>screen-order</i>] [-t <i>token</i>] [-x <i>xinerama</i>]  /opt/SUNWut/bin/utxconfig -e [-d <i>display</i>] [-t <i>token</i>]  /opt/SUNWut/bin/utxconfig [-o] [-f <i>file</i>]  /opt/SUNWut/bin/utxconfig [-i] [-f <i>file</i>] </pre>
<b>DESCRIPTION</b>	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.
<b>OPTIONS</b>	<p>The following options are supported.</p> <ul style="list-style-type: none"> <li>-a                Allows the setting or listing of the default values. Only superuser may change the default settings.</li> <li>-c <i>config-file</i>   Sets a specific <i>configfile</i> to use. The usage of this option is beyond the scope of this manual.</li> <li>-d <i>display</i>        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.</li> <li>-D                Debug flag.</li> <li>-e                Erases all specific settings for the session. All settings return to their default values.</li> <li>-f <i>file</i>           Specifies a file to be used in conjunction with -o or -i.</li> <li>-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.</li> <li>-l                Lists out the current settings for the session. If no specific values have been set for the session, the default values are printed.</li> <li>-L <i>geom</i>           Lists out the X server screen device start-up arguments for the user preferred geometry set with -R or for <i>geom</i> if none is set. The use of this option is beyond the scope of this manual.</li> </ul>

- 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 *WIDTHxHEIGHT*, 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 *COLSxROWS*. 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** utxlock - Sun Ray utility for locking a window session.

**SYNOPSIS** /opt/SUNWut/bin/utxlock

**DESCRIPTION** The `utxlock` utility locks the current display in a windowing-environment specific manner. If the current environment is Gnome, `xscreensaver-command` is utilized. If the current environment is CDE, `dtsession` is utilized. Otherwise `xlock` will be utilized. A user may disable any screen lock behavior by setting the environment variable `SUN_SUNRAY_UTXLOCK_PREF` to `NULL`. Any other value will be used as a command line to use for the invoking a screen lock command instead of the default behavior.

SRSS will invoke `utxlock` on any session disconnect. If users wish to disable this behavior they should add the following line to their `$HOME/.dtprofile`:  
`SUN_SUNRAY_UTXLOCK_PREF=; export SUN_SUNRAY_UTXLOCK_PREF`

As another example, if a user had their own screenlock program called `mylock`, and they wanted to pass it the argument `-l`, they could use it by adding the following line to their `$HOME/.dtprofile`: `SUN_SUNRAY_UTXLOCK_PREF="$HOME/bin/mylock" -l; export SUN_SUNRAY_UTXLOCK_PREF`

**OPTIONS** No options are supported.

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

Attribute Type	Attribute Value
Availability	SUNWutu





- 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<sup>2</sup>. 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<sup>2</sup> when a threshold velocity of *thresh* pixels/second is exceeded.

