

Sun Ray Software

Administration Guide for Release 5.4



E35309-04
June 2013

Sun Ray Software: Administration Guide for Release 5.4

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Oracle Virtual Desktop Client software is an included component of Oracle's Sun Ray Software and Oracle Virtual Desktop Infrastructure software products that must be separately downloaded from Oracle Software Delivery Cloud (<https://edelivery.oracle.com>). Use of Oracle Virtual Desktop Client is subject to the Oracle software license agreement provided with and/or applying to Sun Ray Software and Oracle Virtual Desktop Infrastructure.

Abstract

This guide describes how to install, configure, and manage the Sun Ray Software 5.4 release.

Document generated on: 2013-06-27 (revision: 2528)

Table of Contents

Preface	xiii
1. Overview	1
1.1. What is Sun Ray Computing?	1
1.1.1. Stateless	1
1.1.2. Secure	1
1.1.3. Available	2
1.2. Parts of the Sun Ray Environment	2
1.2.1. Desktop Clients	3
1.2.2. Physical Network	4
1.2.3. Sun Ray Server	4
1.2.4. Desktop Environments	4
1.3. Management Areas	5
2. Planning a Sun Ray Network Environment	7
2.1. Using a Shared Network Configuration	7
2.1.1. Configuring a Shared Network	8
2.1.2. VPN Capability	8
2.1.3. IP MultiPathing (Oracle Solaris 10)	9
2.1.4. IPv4 and IPv6	9
2.1.5. Network Performance Considerations	9
2.2. Configuring Sun Ray Server Discovery	9
2.2.1. Firmware Server	10
2.2.2. Session Server	11
2.2.3. Using Domain Name Service (DNS)	11
3. Installing and Configuring	13
3.1. Product Requirements	13
3.1.1. Operating System Requirements	14
3.1.2. Sun Ray Operating Software	14
3.1.3. Windows Remote Desktop Support	14
3.1.4. Feature Differences Between Oracle Solaris and Oracle Linux Platforms	15
3.1.5. Differences Between Oracle Solaris 10 and Oracle Solaris 11 Platforms	16
3.1.6. Disk Space Requirements	16
3.1.7. Oracle Solaris 10 Prerequisites	16
3.1.8. Oracle Solaris 11 Prerequisites	17
3.1.9. Oracle Linux Prerequisites	18
3.1.10. Java Runtime Environment (JRE) Requirements	21
3.1.11. Sun Ray Admin GUI Web Server Requirements	21
3.1.12. Sun Ray Admin GUI Web Browser Requirements	22
3.1.13. Sun Ray Data Store Port Requirements	22
3.1.14. Ports and Protocols	23
3.2. Installing	26
3.2.1. Using the <code>utsetup</code> Command	26
3.2.2. Not Using the <code>utsetup</code> Command	26
3.2.3. Automating Sun Ray Software Installations	27
3.2.4. Installing Firmware Before Sun Ray Software Installation	27
3.2.5. How to Install Sun Ray Software	28
3.2.6. Post-Installation Configuration	30
3.2.7. How to Install the Windows Connector Components on a Windows System	31
3.2.8. How to Clone a Sun Ray Server	34
3.2.9. How to Install and Configure a Sun Ray Server With Default Settings	36
3.2.10. How to List the Current Sun Ray Software Version	37
3.2.11. How to Remove Sun Ray Software	37

3.2.12. Installation (utinstall) Error Messages	37
3.3. Configuring Oracle Solaris 11 Trusted Extensions	39
3.3.1. How to Configure Sun Ray Software on Oracle Solaris 11 Trusted Extensions	39
3.4. Configuring Oracle Solaris 10 Trusted Extensions	41
3.4.1. How to Configure a Private Network on Oracle Solaris 10 Trusted Extensions	41
3.4.2. How to Configure Shared Multilevel Ports (MLP) for Sun Ray Services	42
3.4.3. How to Increase the Number of X Server Ports	43
3.4.4. How to Configure the Windows Connector on Oracle Solaris Trusted Extensions	43
3.5. Upgrading	45
3.5.1. Installing Firmware Before Sun Ray Software Upgrade	45
3.5.2. How to Upgrade Sun Ray Software	45
3.5.3. Planning Upgrades Using Failover Groups	49
3.5.4. How to Preserve Sun Ray Software Configuration Data	50
4. Admin GUI and Commands	53
4.1. Sun Ray Software Commands	53
4.1.1. How to Set Up Access to the Sun Ray Software Man Pages	56
4.2. Administration Tool (Admin GUI)	56
4.2.1. Administrative Name and Password	57
4.2.2. Admin GUI Tab Descriptions	57
4.2.3. How to Log In to the Administration Tool (Admin GUI)	59
4.2.4. How to Change the Admin GUI Locale	60
4.2.5. How to Change the Admin GUI to English Locale	60
4.2.6. How to Change the Admin GUI Timeout	60
4.2.7. How to Enable or Disable Multiple Administration Accounts (Oracle Linux)	61
4.2.8. How to Enable or Disable Multiple Administration Accounts (Oracle Solaris 11)	62
4.2.9. How to Enable or Disable Multiple Administration Accounts (Oracle Solaris 10)	62
4.2.10. How to Audit Admin GUI Sessions	63
5. Sun Ray Server and Networking	65
5.1. Log Files	65
5.2. How to Start or Stop Sun Ray Services	67
5.2.1. How to Stop Sun Ray Services	67
5.2.2. How to Start Sun Ray Services (Warm Restart)	67
5.2.3. How to Start Sun Ray Services (Cold Restart)	67
5.3. How to Check and Fix Corrupted Configuration Files (Oracle Solaris 10)	68
5.4. How to Unconfigure a Sun Ray Server	69
5.5. How to Disconnect a Sun Ray Server From the Interconnect	69
5.6. User Fields in the Sun Ray Data Store	70
5.7. Network Troubleshooting	70
5.7.1. Network Load	70
5.7.2. The <code>utcapture</code> Utility	70
5.7.3. <code>utcapture</code> Examples	71
5.7.4. The <code>utquery</code> Command	71
6. Failover Groups	73
6.1. Failover Groups Overview	73
6.2. Failover Process	74
6.3. Load Balancing	74
6.4. Mixing Different Sun Ray Servers	74
6.5. Authentication Requirements	75
6.6. Dedicated Primary Servers for Data Store	75
6.7. Setting Up a Failover Group	75
6.7.1. How to Configure a Primary Server	76
6.7.2. How to Add a Secondary Server	76
6.7.3. How to Synchronize Primary and Secondary Sun Ray Servers	77
6.7.4. How to Change the Group Manager Signature	77

6.8. Additional Failover Group Tasks	78
6.8.1. How to Take a Server Offline and Online	78
6.8.2. How to Disable Load Balancing	78
6.8.3. How to Show the Current Sun Ray Data Store Replication Configuration	78
6.8.4. How to Remove the Replication Configuration	78
6.8.5. How to View the Failover Group Status	78
6.9. Recovery Issues and Procedures	79
6.9.1. How to Rebuild the Primary Server's Administration Data Store	79
6.9.2. How to Replace the Primary Server with a Secondary Server	80
6.9.3. Secondary Server Recovery	80
6.10. Group Manager Details	80
6.10.1. Group Manager Configuration	81
7. Sessions and Tokens	83
7.1. Sessions Overview	83
7.1.1. Authentication Manager	84
7.1.2. Session Manager	85
7.2. Managing Sessions	86
7.2.1. How to Redirect a Session	86
7.2.2. How to Disconnect a Session	87
7.2.3. How to Terminate a Session	88
7.2.4. How To Identify a Hung Session	88
7.2.5. How To Kill a Hung Session	88
7.3. Tokens	88
7.3.1. Registering Tokens	89
7.3.2. How to Register a Token	89
7.3.3. How to Register a Pseudo-Token	89
7.3.4. How to Enable, Disable, or Delete a Token	90
7.4. Token Readers	90
7.4.1. How to Configure a Token Reader	90
7.4.2. How to Locate a Token Reader	91
7.4.3. How to Get a Token ID From a Token Reader	91
7.5. Session Troubleshooting	92
7.5.1. Problem: The <code>dtlogin</code> daemon cannot start the Xsun or Xnewt server properly.	92
8. Smart Card Services	93
8.1. Overview	93
8.2. Smart Card Bus Protocol	94
8.3. Smart Card Configuration Files	94
8.4. Smart Card Probe Order	94
8.5. Hotdesking with Smart Cards	95
8.6. Configuring Smart Card Services	95
8.6.1. How to Configure Primary Smart Card Readers for Hotdesking and Authentication	95
8.6.2. How to Configure External CCID-Compliant USB Smart Card Readers for Authentication (Oracle Solaris)	96
8.6.3. How to Add a Smart Card Configuration File	96
8.6.4. How to Change the Smart Card Probe Order	97
8.6.5. How to Change the Smart Card Bus Protocol (Oracle Solaris)	97
8.7. Troubleshooting Smart Card Services	98
8.7.1. Smart Card Transaction Problems	98
8.8. CCID IFD Handler for External USB Smart Card Readers (Oracle Solaris)	99
8.8.1. How to Install CCID IFD Handler	99
8.8.2. How to Uninstall CCID IFD Handler	99
8.8.3. Known Issues	100
9. Hotdesking	101
9.1. Hotdesking Overview	101

9.2. Hotdesking Without Smart Cards	101
9.2.1. NSCM and Failover Groups	102
9.2.2. How to Enable NSCM Sessions	102
9.2.3. How to Log in to an NSCM Session	103
9.3. Regional Hotdesking	104
9.3.1. Regional Hotdesking Process	105
9.3.2. Regional Hotdesking Site Requirements	105
9.3.3. Providing Site Integration Logic	105
9.3.4. How to Configure a Site-specific Mapping Library	105
9.3.5. How to Use Token Readers with Regional Hotdesking	106
9.3.6. How to Configure the Sample Data Store	106
9.4. Remote Hotdesk Authentication (RHA)	107
9.4.1. How to Disable Remote Hotdesk Authentication	108
9.4.2. How to Re-enable Remote Hotdesk Authentication	108
10. Kiosk Mode	109
10.1. Kiosk Overview	109
10.2. Kiosk Mode Security and Failover Considerations	110
10.3. Kiosk User Accounts	110
10.3.1. Characteristics	110
10.3.2. Restrictions and Safe Guards	111
10.3.3. Administering the Kiosk User Pool	111
10.4. Session Type Components	111
10.4.1. Session Descriptor	111
10.4.2. Session Script	112
10.5. How to Configure Kiosk Mode and User Accounts	112
10.6. How to Add Kiosk User Accounts	112
10.7. How to Configure a Kiosk Mode Session Type	113
10.8. How to Enable and Disable Kiosk Mode	116
10.8.1. Unconfiguring Kiosk Mode Disables Kiosk Policy	117
10.9. How to Override the Default Kiosk Mode Policy	117
10.10. Configuring the Windows Connector Kiosk Session Type	119
10.10.1. How to Configure a Kiosk Mode Session Type for the Windows Connector	120
10.11. Configuring the VMware View Connector Kiosk Session Type	122
11. Client-Server Security	123
11.1. Client-Server Security Overview	123
11.2. Encryption and Authentication	124
11.2.1. Security Modes	124
11.2.2. How to Force Encryption	125
11.2.3. How to Force Server Authentication	125
11.2.4. How to Disable Client Authentication	126
11.2.5. How to Force Client Authentication From All Clients	126
11.3. Managing Client Keys	127
11.3.1. Key Fingerprint	128
11.3.2. How to Deny Access to Clients With Unconfirmed Keys	128
11.3.3. How to Confirm a Specific Client Key	128
11.3.4. How to Confirm All Unconfirmed Client Keys	129
11.3.5. How to Display a Client's Fingerprint Key from a Sun Ray Client	129
11.3.6. How to Display All Client Keys	130
11.3.7. How to Display All Keys for a Specific Client	130
11.3.8. How to Delete a Specific Client Key	131
11.3.9. How to Delete All Client Keys for a Specific Client	131
11.4. Displaying Security Status	131
11.4.1. How to Display Security Status for a Sun Ray Client	131
11.4.2. How to Display Security Status for All Sessions	131

11.5. Authentication Troubleshooting	132
11.5.1. Error Messages	132
12. Multiple Monitor Configurations	135
12.1. Multi-Monitor	135
12.1.1. How to Set a Sun Ray Client's Multi-Monitor Configuration With Optimal Settings	136
12.1.2. How to Set a Sun Ray Client's Multi-Monitor Configuration With Customized Settings	136
12.2. Multihead Groups	138
12.2.1. Creating a Multihead Group	138
12.2.2. Multihead Group Screen Indicator	139
12.2.3. Creating a Single Screen Across Several Monitors (Xinerama)	139
12.2.4. How to Create a New Multihead Group	139
12.2.5. How to Enable the Multihead Group Policy	140
12.2.6. How to Manually Set Multihead Group Screen Dimensions	141
12.2.7. How to Manually Set Multihead Group Geometry	141
12.2.8. How to Disable Multihead Group for a Session	142
12.2.9. How to Enable and Disable Xinerama	142
12.2.10. How to Disconnect a Secondary Client	142
13. Desktop Clients	145
13.1. Managing Desktop Clients	146
13.1.1. Client ID Differences Between Oracle Virtual Desktop Clients and Sun Ray Clients	146
13.1.2. Dynamic Session Resizing	147
13.1.3. How to List Available Sun Ray Servers	150
13.1.4. How to List the Available Clients	150
13.1.5. How to Display Sun Ray Client Information	150
13.1.6. How to Configure a Client's Location and Information	151
13.1.7. Audio Output Troubleshooting (Oracle Solaris 10 and Oracle Linux 5)	151
13.1.8. Audio Output Troubleshooting (Oracle Solaris 11 and Oracle Linux 6)	153
13.2. Sun Ray Clients	155
13.2.1. How to Centralize Sun Ray Client Configurations (.parms)	155
13.2.2. Sun Ray Client Hot Keys	158
13.2.3. How to Change Sun Ray Client Audio and Display Settings (Sun Ray Settings GUI)	160
13.2.4. How to Modify Screen Resolutions	161
13.2.5. How to Power Cycle a Sun Ray Client	162
13.2.6. How to Enable or Disable XRender	162
13.2.7. How to Configure Screen Rotation	163
13.2.8. How to Disable Screen Blanking on a Sun Ray Client	163
13.2.9. How to Enable the NumLock Key for All Sun Ray Sessions	165
13.2.10. Keyboard Country Codes	166
13.2.11. Sun Ray Client Boot Process	167
13.3. Oracle Virtual Desktop Clients	170
13.3.1. Oracle Virtual Desktop Clients Overview	170
13.3.2. Using External Devices on the Client Computer	171
13.3.3. How to Enable Access for Oracle Virtual Desktop Clients	171
13.3.4. How to Enable the Clipboard Service for Oracle Virtual Desktop Clients	173
13.3.5. Oracle Virtual Desktop Client Troubleshooting	173
14. Sun Ray Client Firmware	177
14.1. Firmware Overview	177
14.2. Firmware Server Discovery	178
14.3. How to Update Firmware on Sun Ray Clients	178
14.4. How to Enable and Disable the Configuration GUI on All Sun Ray Clients	179
14.5. How to Modify a Sun Ray Client's Local Configuration (Configuration GUI)	181

14.5.1. Security Configuration Repository	181
14.5.2. Configuration GUI Menu Descriptions	182
14.5.3. How to Load a Remote Configuration File	187
14.6. VPN Support	189
14.6.1. How to Configure VPN Using Cisco Hybrid Authentication	190
14.7. IPsec	190
14.8. 802.1x Authentication	191
14.8.1. How to Configure and Enable 802.1x Authentication on a Sun Ray Client	191
14.9. How to Display Firmware Versions for All Currently Connected Sun Ray Clients	193
14.10. How to Display the Firmware Version from a Sun Ray Client	193
14.11. How to Synchronize the Sun Ray Client Firmware	193
14.12. How to Downgrade Firmware on a Sun Ray Client	193
14.13. How to Disable All Sun Ray Client Firmware Updates	195
15. Peripherals	197
15.1. Peripherals Overview	197
15.2. Enabling and Disabling Device Services	198
15.2.1. How to Determine the Current State of Device Services	198
15.2.2. How to Enable or Disable USB Device Services	199
15.3. Device Availability Per Session	199
15.4. Accessing Serial Devices and USB Printers	199
15.4.1. Device Links	199
15.4.2. Device Nodes	200
15.4.3. Device Node Ownership	201
15.4.4. Hotdesking and Device Node Ownership	201
15.4.5. Setting Up Serial Devices	201
15.4.6. Setting Up USB Printers	201
15.5. Accessing USB Mass Storage Devices	204
15.5.1. Device Nodes and Links (Oracle Solaris)	204
15.5.2. Device Nodes and Links (Oracle Linux)	204
15.5.3. Mount Points	204
15.5.4. Device Ownership and Hotdesking	205
15.5.5. Mass Storage Devices and Idle Sessions	205
15.5.6. Commands for Common Disk Operation (Oracle Solaris)	205
15.5.7. Commands for Common Disk Operation (Oracle Linux)	206
15.5.8. How to Unmount a Mass Storage Device From a Client	206
15.5.9. Troubleshooting Mass Storage Devices	207
15.6. USB Headsets	207
15.6.1. Tested USB Headsets	207
15.6.2. Tested Applications	208
15.6.3. Additional Notes	208
15.7. USB Device Operations Failing After Idle Timeout Limit	208
16. Troubleshooting Icons	211
16.1. On-Screen Display (OSD) Icons	211
16.2. Server Policy Icons	212
16.3. Troubleshooting Icon Quick Reference	213
16.4. DHCP State Codes	215
16.5. Encryption and Authentication States	216
16.6. Power LEDs	216
16.7. (1) Sun Ray Client Startup Icon	217
16.8. (2) Firmware Download in Progress Icon	217
16.9. (3) Updating Firmware Icon	218
16.10. (4) Firmware Download Diagnostics Icon	218
16.11. (11-14) Network Status Icons	219
16.12. (15) Session Refused Icon	220

16.13. (16) Bus Busy Icon	220
16.14. (20) 802.1x Authentication Icon	221
16.15. (21) Network Connection Verified Icon	221
16.16. (22) Waiting to Connect to Authentication Manager Icon	222
16.17. (23) No Ethernet Signal Icon	223
16.18. (25) Redirection Icon	224
16.19. (26) Wait for Session Icon	224
16.20. (27) DHCP Broadcast Failure Icon	225
16.21. (28) Establishing VPN Connection Icon	225
16.22. (29) VPN Connection Established Icon	226
16.23. (30) VPN Connection Error	226
16.24. (31-34) Network Status Icons	226
16.25. (41-44) Network Status Icons	227
16.26. (46) No Access to Server Icon	228
16.27. (47) No Access for Oracle Virtual Desktop Clients Icon	228
16.28. (48) No Access: Registration Required Icon	228
16.29. (49) No Access: Key Rejected Icon	229
16.30. (50) No Access: Security Policy Violation Icon	229
16.31. (51-54) Network Status Icons	229
16.32. (60) Insert Card Icon	229
16.33. (61) Waiting for Primary Sun Ray Client Icon	230
16.34. (62) Token Reader Icon	230
16.35. (63) Card Error Icon	230
16.36. (64) Waiting For Access Icon	231
17. Windows Connector	233
17.1. Windows Connector Overview	234
17.2. Requirements	236
17.3. Using the Windows Connector	236
17.3.1. How to Start a Windows Session	236
17.3.2. How to Start a Windows Session Within Java Desktop System (Oracle Solaris 10) ...	238
17.3.3. How to Lock a Windows Session	238
17.3.4. How to Set Up Access to the <code>uttsc</code> Command	239
17.3.5. How to Set Up a Desktop Shortcut to Start a Windows Session	239
17.3.6. How to Separate Settings for Session Locale and Keyboard Layout	239
17.4. Audio Input	239
17.4.1. Enabling Audio Input on Windows 7 and Windows Server 2008 R2	240
17.5. Video Acceleration	240
17.5.1. Video Acceleration Requirements	240
17.5.2. Videos Accelerated	241
17.5.3. Audio Accelerated	243
17.5.4. Additional Notes	243
17.5.5. How to Enable Video Redirection on Windows Server 2008 R2	245
17.5.6. Video Acceleration Troubleshooting	245
17.6. USB Device Redirection	250
17.6.1. Device Access	251
17.6.2. Supported Configurations	251
17.6.3. Tested USB Devices	251
17.6.4. Additional Notes	251
17.6.5. How to Add USB Drivers to a Virtual Machine	252
17.6.6. USB Redirection Troubleshooting	253
17.7. Hotdesking	254
17.7.1. Hotdesking Behavior	255
17.7.2. Location Awareness	255
17.8. Session Directory	257

17.9. Network Security	257
17.9.1. Built-in RDP Network Security	258
17.9.2. Enhanced Network Security	258
17.10. Automatic Reconnection	260
17.11. Compression	260
17.11.1. How to Disable Compression	260
17.12. Licensing	260
17.12.1. Per-user Mode Versus Per-device Mode	261
17.13. Smart Cards	262
17.13.1. How to Enable Smart Card Readers on a Windows System	262
17.13.2. How to Set Up Smart Card Login for Windows	262
17.14. Multi-Monitor Support	262
17.15. Dynamic Session Resizing	263
17.16. Printing	263
17.16.1. How to Set Up Print Queues (Oracle Solaris 10)	264
17.16.2. How to Set Up Print Queues (Oracle Linux)	264
17.16.3. How to Make Sun Ray Printers Available to a Windows Session	265
17.16.4. How to Manage Printer Configurations for Users	266
17.16.5. How to Set Up Follow-Me Printing	266
17.16.6. Printers Troubleshooting	267
17.17. Accessing Serial Devices	267
17.18. uttsc Error Messages	268
17.18.1. General Troubleshooting	269
18. VMware View Connector	271
18.1. VMware View Connector Overview	271
18.2. Requirements	271
18.3. Configuring the VMware View Environment	272
18.3.1. How to Disable Secure Tunnel Connections to View Desktops	272
18.3.2. How to Enable Non-SSL Connections to the View Connection Server	272
18.3.3. How to Enable SSL Connections to the View Connection Server	272
18.4. Configuring the VMware View Connector Kiosk Session Type	273
18.4.1. How to Configure a Kiosk Mode Session Type for the VMware View Connector	273
18.5. VMware View Connector Troubleshooting	274
18.5.1. Error Messages	274
18.5.2. Desktop tries to open, but immediately disconnects	275
19. Alternate Network Configurations	277
19.1. Alternate Network Configurations Overview	277
19.2. Updating the Default /etc/hosts File Before Configuring Sun Ray Network (Oracle Linux)	278
19.3. Using a Shared Network Configuration Without External DHCP Services	278
19.3.1. Shared Network Configuration Worksheet	278
19.3.2. How to Configure a Sun Ray Server on a Shared Network to Provide DHCP Services	279
19.3.3. How to List the Current Network Configuration	280
19.3.4. How to Delete a LAN Subnet	280
19.3.5. Example Shared Network Setups	281
19.4. Using a Private Network Configuration	289
19.4.1. Private Network Configuration Worksheet	290
19.4.2. How to Configure a Sun Ray Server in a Private Network	292
19.4.3. How to List the Current Network Configuration	293
19.4.4. How to Print a Private Network Configuration	293
19.4.5. How to Delete an Interface	293
19.4.6. Example Private Network Setup	293
19.5. Sun Ray Client Initialization Requirements Using DHCP	296

19.5.1. DHCP Basics	296
19.5.2. DHCP Parameter Discovery	297
19.5.3. DHCP Relay Agent	298
19.5.4. Simplifying DHCP Configuration of Remote Sun Ray Clients	298
19.5.5. Standard DHCP Parameters	299
19.5.6. Vendor-specific DHCP Options	300
19.5.7. Encapsulated Options	302
19.6. Failover Groups	303
19.6.1. Network Topologies	303
19.6.2. Setting Up IP Addressing	305
19.6.3. Sun Ray Server Failover Group Worksheet	309
20. Performance Tuning	311
20.1. How to Gain Network Performance for Sun Ray 3 Series Clients	311
20.2. How to Improve Network Performance by Disabling CPU Binding (Oracle Solaris 11)	311
20.3. How to Improve Sun Ray Client Performance by Decreasing Buffering on the Network Switch (Oracle Solaris)	311
20.4. Improving Sun Ray Client Start-Up Time by Disabling Spanning Tree Protocol on the Network Switch	312
20.5. Applications	312
20.6. Tuning the Java Desktop System	313
20.7. Excessive Disk Swapping	313
20.8. Screensaver Resource Consumption	313
20.8.1. How to Disable Screensavers (Oracle Solaris 10)	313
A. IPsec Support	315
A.1. Overview	315
A.2. IKE Configuration	315
A.2.1. remote Directive	316
A.2.2. sainfo Directive	318
A.2.3. Example IKE Configuration Files	318
A.3. IPsec Configuration GUI Menu	319
A.4. IPsec Configuration Examples	319
A.4.1. Oracle Linux 5 Pre-Shared Key	320
A.4.2. Oracle Linux 5 Certificates	321
A.4.3. Oracle Linux 6 Pre-Shared Key	322
A.4.4. Oracle Linux 6 Certificates	323
A.4.5. Oracle Solaris Pre-Shared Key	324
A.4.6. Oracle Solaris Certificates	326
A.4.7. Sun Ray Client Configuration	327
A.4.8. IPsec Verification	328
B. Admin GUI Help	329
B.1. Servers Tab	329
B.1.1. Server Details	330
B.2. Sessions Tab	331
B.3. Desktop Units Tab	333
B.3.1. Desktop Units Properties	334
B.4. Tokens Tab	335
B.4.1. Token Properties	337
B.5. Advanced Tab	338
B.5.1. Security	338
B.5.2. System Policy	339
B.5.3. Kiosk Mode	341
B.5.4. Card Probe Order	342
B.5.5. Data Store Password	342
B.6. Log Files Tab	343

C. Third Party Licenses	345
Glossary	367

Preface

This document provides information for the Sun Ray Software 5.4 product.

Audience

This document is intended for users with system administration experience. It is assumed that readers are familiar with Web technologies and have a general understanding of Windows and UNIX platforms.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Related Documents

The entire set of documentation for this product is available at:

<http://www.oracle.com/technetwork/server-storage/sunrayproducts/docs>

The documentation set includes the following manuals:

- *Sun Ray Software 5.4 Administration Guide*
- *Sun Ray Software 5.4 Release Notes*
- *Sun Ray Software 5.4 Security Guide*
- *Oracle Enterprise Manager System Monitoring Plug-in Installation Guide for Sun Ray Software*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
<code>monospace</code>	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Chapter 1. Overview

Table of Contents

1.1. What is Sun Ray Computing?	1
1.1.1. Stateless	1
1.1.2. Secure	1
1.1.3. Available	2
1.2. Parts of the Sun Ray Environment	2
1.2.1. Desktop Clients	3
1.2.2. Physical Network	4
1.2.3. Sun Ray Server	4
1.2.4. Desktop Environments	4
1.3. Management Areas	5

This chapter provides an overview of the Sun Ray technology and describes the major areas and features of the Sun Ray environment.

1.1. What is Sun Ray Computing?

Sun Ray computing is a thin client implementation that offers both desktop-like user functionality and sufficient speed and reliability for mission-critical applications. Sun Ray Software supports both hardware and software-based clients and runs on both Oracle Linux and Oracle Solaris, including Oracle Solaris Trusted Extensions.

Other client-server models typically use combinations of remote and local operating systems, applications, memory, and storage, but the Sun Ray computing model moves all computing to a server. Instead of running applications, storing data, and doing computation on a desktop device like a PC, the Sun Ray model simply passes input and output data between clients and the Sun Ray server, where the operating system and applications are located.

The following descriptions of the Sun Ray Computing attributes are based on the Sun Ray Clients, although many of the points also apply to Oracle Virtual Desktop Clients.

1.1.1. Stateless

Sun Ray Clients have no local disks, applications, or operating systems and are therefore considered *stateless*. This setup is what makes them true thin clients. Stateless devices are inexpensive to maintain because they do not require administrators or technicians to install, upgrade, or configure software or to replace mechanical components on the desktop.

A Sun Ray Client contains only a firmware module that performs a small set of tasks: it sends keyboard and mouse events and displays pixel data. If a desktop device contains an operating system that can execute code at the request of a user, it has *state* and it is not a true thin client. This type of device requires updating and maintenance at the desktop rather than server level and it is susceptible to viruses.

1.1.2. Secure

Sun Ray Clients are also extremely secure. For instance, managing USB mass storage devices, that is, controlling the ability to enable or disable their use, is done at the server or group level. This ability enables

sites with particular security or intellectual property concerns to eliminate many of the risks imposed by PCs and other fat clients, which rely on local operating systems, local applications, and local data caches. Critical data can be compromised or lost when the physical device hosting the "fat" client is stolen or damaged.

1.1.3. Available

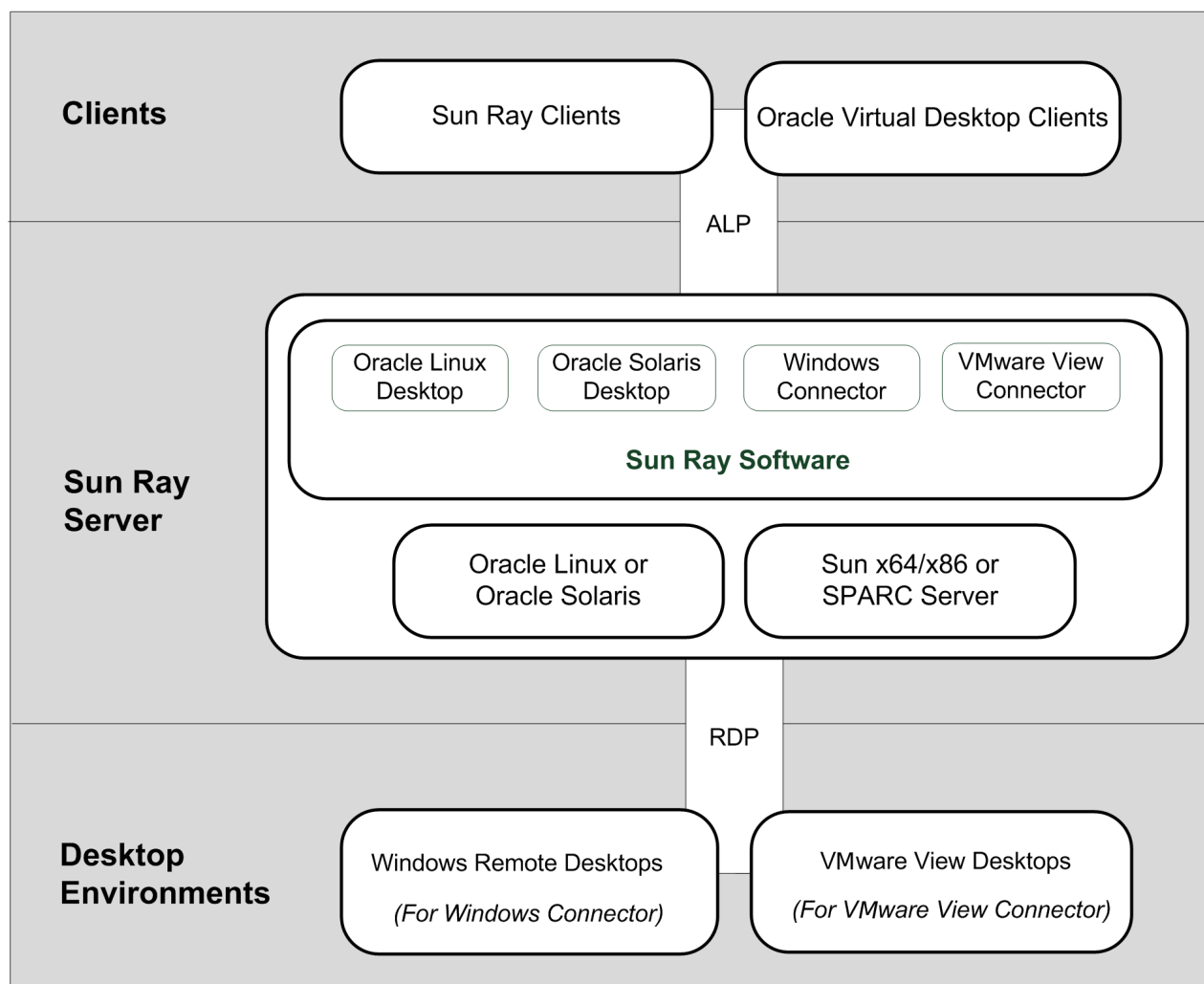
A Sun Ray session is a group of services controlled by a session manager and associated with a user through an authentication token. The sessions reside on a server rather than on the desktop. Because Sun Ray Clients are stateless, a session can be directed or redirected to any Sun Ray Client on the appropriate network or subnetwork when a user logs in or inserts a smart card.

Although the session continues to reside on a server, the session appears to follow the user to the new client. This functionality, called hotdesking, provides the ability of users to access their sessions from any client on their network. Hotdesking can be implemented with smart cards or without smart cards through the non-smart card session mobility (NSCM) feature.

Most large Sun Ray implementations also include one failover group of Sun Ray servers to ensure uninterrupted service whenever a server is off-line. When a failover group is configured, the Sun Ray Software optimizes performance by spreading the computing load among the servers in the group.

1.2. Parts of the Sun Ray Environment

The Sun Ray Software is the central piece of the overall Sun Ray environment. From a high-level perspective, a Sun Ray environment consists of three main areas: the clients, the physical network, and the Sun Ray server, where the Sun Ray Software is installed. [Figure 1.1, "Parts of the Sun Ray Environment"](#) shows the high-level connections within a Sun Ray environment.

Figure 1.1. Parts of the Sun Ray Environment

The following sections provide a high-level overview of each area.

1.2.1. Desktop Clients

Currently, there are two main types of desktop clients: a Sun Ray Client and an Oracle Virtual Desktop Client.

1.2.1.1. Sun Ray Client

A Sun Ray Client is a hardware unit that can potentially exceed the full functionality of a desktop computer, but with less administrative and environmental costs. A Sun Ray Client acts as a frame buffer on the client side of the network. Applications run on a Sun Ray server and render their output to a virtual frame buffer. The Sun Ray Software formats the rendered output and sends it to the appropriate Sun Ray Client, where the output is interpreted and displayed.

From the point of view of network servers, Sun Ray Clients are identical except for their Ethernet mac address. If a Sun Ray Client ever fails, it can be easily replaced because no data resides on it. An IP address is leased to each Sun Ray Client when it is connected and it can be reused when the client is disconnected. IP address leasing is managed by DHCP.

Various models of Sun Ray Clients are available, differing primarily with respect to size, type, and supported monitor resolution. However, all Sun Ray Clients include a smart card reader, a keyboard, and a mouse.

For the smart card reader, the industry standard PC/SC-lite API is included for developers who want to encode custom applications or other information in their users' smart cards.

By default, a Sun Ray Client uses the same Oracle Solaris or Oracle Linux operating system as the associated Sun Ray server, known as a regular Sun Ray session. However, the Windows and VMware View connectors enable users to access a remote Windows desktop session or a VMware View session on the client. By configuring kiosk mode, users can bypass the regular Oracle Solaris or Oracle Linux Sun Ray session altogether and be taken directly to the assigned connector session.

1.2.1.2. Oracle Virtual Desktop Client

An Oracle Virtual Desktop Client is a software version of a Sun Ray Client. The Oracle Virtual Desktop Client application runs on an ordinary computer or tablet and provides access to a Sun Ray session. It is supported and can be installed on Windows, Linux, Mac OS X, iPad, or Android. An Oracle Virtual Desktop Client supports most of the standard Sun Ray Client functionality.

1.2.2. Physical Network

As with most networked environments, the Sun Ray environment needs a well-designed network, and it can be configured in one of several ways.

For detailed descriptions of the network configuration types and instructions on configuring each network type, see [Chapter 2, *Planning a Sun Ray Network Environment*](#).

1.2.3. Sun Ray Server

The Sun Ray server runs the Sun Ray Software and it is the foundation of the Sun Ray environment. It provides all the necessary administrative support for Sun Ray Clients and Oracle Virtual Desktop Clients.

The important first step is determining what operating system environment your users need, and then you can set up your Sun Ray environment accordingly. If your users need an Oracle Linux environment, you can install Oracle Linux on the Sun Ray servers. Vice versa for users who need an Oracle Solaris environment. If your users only plan to use a remote Windows desktop session, you can choose either Oracle Linux or Oracle Solaris on the Sun Ray server, since the underlying desktop will be hidden from the users. There are a [few differences](#) when using Oracle Linux versus the Oracle Solaris operating system for the Sun Ray server, so take that into consideration.

When providing your users with a Windows environment, it is also important to understand how the Sun Ray Software features are supported on different versions of the Windows operating systems in a remote desktop environment. Although most features are provided for all of the supported Windows platforms, there are differences in how they are implemented.

To communicate with the desktop clients, the Sun Ray server uses the Appliance Link Protocol (ALP), which is a suite of network protocols that are automatically available as part of the Sun Ray Software. The Sun Ray server also uses the standard Microsoft Remote Desktop Protocol (RDP) to access the remote Windows desktops when used.

1.2.4. Desktop Environments

When providing access to remote Windows desktops, the Windows systems must be accessible to the Sun Ray server and the Sun Ray network. Then, you can configure the Windows connector or VMware View connectors for users to access the remote Windows desktops, including using kiosk mode to automatically provide the Windows desktop when the user logs in to the desktop client.

1.3. Management Areas

Table 1.1, "Sun Ray Software Management Areas" provides a brief introduction to all the areas in the Sun Ray Software that require management and administration. Every item on this list has a corresponding chapter that provides detailed information.

Table 1.1. Sun Ray Software Management Areas

Management Area	Description
CLI and Admin GUI	Sun Ray Software has both a command-line interface (CLI) and an Admin GUI for administrative functions. The GUI presents a clear view of administrative functions, with a tab-based navigational model and context-sensitive help.
Sun Ray Server	The Sun Ray server runs the Sun Ray Software and provides sessions to clients.
Failover Group	A failover group consists of two or more servers that provide users with a high level of availability in case one server becomes unavailable.
Hotdesking	<p>Hotdesking, or session mobility, is the ability for a session to "follow" a user between clients. This enables the user to have instantaneous access to the user's windowing environment and current applications from multiple clients. Hotdesking can be implemented with smart cards or without smart cards through the non-smart card session mobility (NSCM) feature.</p> <p>Regional hotdesking promotes hotdesking among server groups, letting users access their sessions across a wider domain.</p>
Kiosk Mode	A way to provide an unlimited variety of desktops or applications to users, even though the actual desktop or application may be running elsewhere. Kiosk mode bypasses the normal authentication methods of the platform and runs anything that the administrator defines. It is the primary way to provide user's access to the Windows connector or VMware View Manager connector sessions without the user ever seeing the client's default desktop.
Client-Server Security	<p>Sun Ray Software provides the ability to manage the security and authentication policies between the desktop clients and the Sun Ray server.</p> <p>For more information about all the security aspects of Sun Ray Software, refer to Sun Ray Software Security Guide.</p>
Multiple Monitor Configurations	<p>Sun Ray Software enables you to merge and control multiple Sun Ray Client screens or heads, using a single keyboard and mouse attached to a primary client. This functionality is important for users who need to monitor many applications or systems simultaneously or to accommodate a single application, such as a large spreadsheet, across multiple screens. To use multiple screens, the administrator sets up multihead groups, consisting of two or more clients.</p> <p>There is also multi-monitor support to provide a single desktop for Sun Ray Clients with dual video connectors.</p>
Desktop Clients	<p>A Sun Ray Client is a hardware unit that can potentially exceed the full functionality of a desktop computer, but with less administrative and environmental costs.</p> <p>An Oracle Virtual Desktop Client is a software application that runs on a common client operating system and provides the ability to connect to a desktop session running on a Sun Ray server, just like a physical Sun Ray Client.</p>

Management Area	Description
Sun Ray Client Firmware	A small firmware module in each Sun Ray Client is managed from the Sun Ray server. The firmware module checks the hardware with a power-on self test (POST) and initializes the client. Maintaining the latest firmware on Sun Ray Clients is an important part of administering the Sun Ray environment. If the GUI mode is enabled on the firmware, the user can also modify a Sun Ray Client's local configuration through a tool called the Configuration GUI.
Windows Connector	Desktop clients can access a remote Windows desktop from a Windows system. Windows support on a Sun Ray Client includes video acceleration and access to USB devices.
VMware View Connector	Desktop clients can access virtual Windows desktops provided through the VMware View Manager.
Citrix XenDesktop Connector	Desktop clients can access virtual Windows desktops provided through the Citrix XenDesktop Web Interface Server. The Citrix XenDesktop connector is not provided with Sun Ray Software. You must download and install the Citrix XenDesktop connector separately. See the Sun Ray Connector for Citrix XenDesktop Administration Guide for details.

Chapter 2. Planning a Sun Ray Network Environment

Table of Contents

2.1. Using a Shared Network Configuration	7
2.1.1. Configuring a Shared Network	8
2.1.2. VPN Capability	8
2.1.3. IP MultiPathing (Oracle Solaris 10)	9
2.1.4. IPv4 and IPv6	9
2.1.5. Network Performance Considerations	9
2.2. Configuring Sun Ray Server Discovery	9
2.2.1. Firmware Server	10
2.2.2. Session Server	11
2.2.3. Using Domain Name Service (DNS)	11

This chapter provides detailed information about setting up the network environment to support Sun Ray Clients and Oracle Virtual Desktop Clients. There are many options for configuring the network to support both firmware provisioning and Sun Ray server discovery, but the recommended approach is a shared network configuration, detailed below.

For other network configurations that are far less common and require advanced configuration of the Sun Ray server and network, see [Chapter 19, Alternate Network Configurations](#).

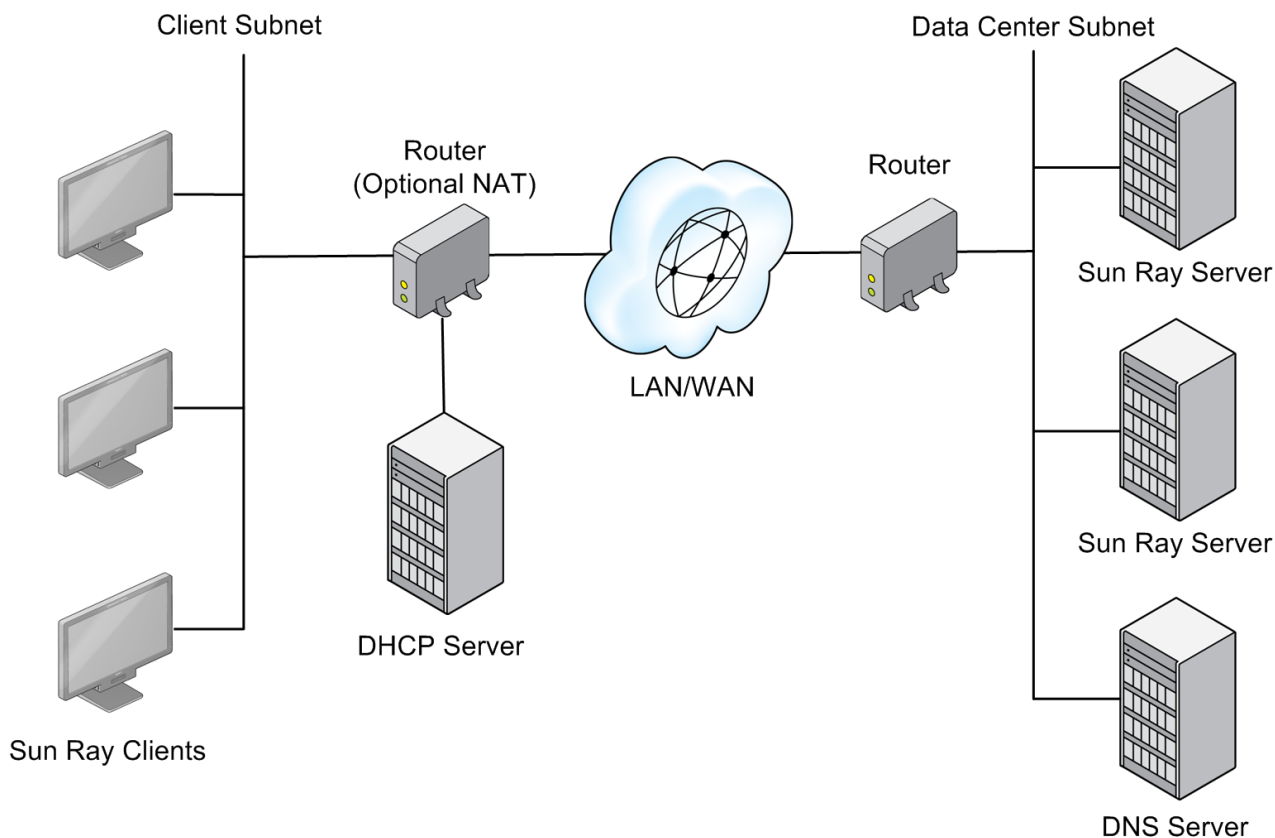
2.1. Using a Shared Network Configuration

By supporting various network configurations, Sun Ray Clients and Oracle Virtual Desktop Clients can be deployed virtually anywhere, subject only to a sufficient quality of network service between the clients and the Sun Ray server. The most common and recommended configuration is a shared network, where the Sun Ray server and clients are part of a Local Area Network (LAN) or Wide Area Network (WAN) and where network services such as DHCP and DNS are already provided by existing servers. The default installation and configuration procedures in this document target this configuration.

A client subnet in a typical shared network configuration meets the following criteria:

- DHCP services are available for Sun Ray Client's IP/network configuration (Sun Ray Clients can be individually configured for static addressing using the Configuration GUI on the Sun Ray Client).
- DNS services are available to resolve the Sun Ray servers providing firmware ([sunray-config-servers](#)) and the Sun Ray servers providing sessions ([sunray-servers](#)).
- The subnet must be able to route to the Sun Ray servers.
- Sun Ray Clients can be behind a Network Address Translation (NAT) device, such as a router or firewall that implements NAT.
- Sun Ray servers must not be behind a NAT device, such as a router or firewall that implements NAT.
- A Sun Ray server requires both a fixed host name and a static IP address. A Sun Ray server cannot be a DHCP client.

[Figure 2.1, "Shared Network Configuration Example"](#) shows an example of using a shared network for a Sun Ray environment.

Figure 2.1. Shared Network Configuration Example**Note**

Given the topology, Sun Ray traffic on shared networks is potentially exposed to an eavesdropper. Modern switched network infrastructures are far less susceptible to snooping activity than earlier shared technologies, but to obtain additional security the administrator may choose to activate the client's encryption and authentication features. These capabilities are discussed in [Chapter 11, Client-Server Security](#).

2.1.1. Configuring a Shared Network

If you use the `utsetup` command for the installation, you are asked to configure the Sun Ray Software to support a shared network with external DHCP/DNS services.

```
Do you want to enable LAN access for Sun Ray clients at this time?
```

If you accept, the `utsetup` command runs the `utadm -L on` command to configure a shared network. See [Section 3.2.1, "Using the utsetup Command"](#) for more information.

2.1.2. VPN Capability

Sun Ray Clients are able to provide a VPN solution for remote users. The IPsec capability in the Sun Ray Client firmware enables the Sun Ray Client to act as a VPN endpoint device. The most commonly used encryption, authentication, and key exchange mechanisms are supported, along with Cisco extensions that enable a Sun Ray Client to interoperate with Cisco gateways that support the Cisco [EzVPN](#) protocol. Sun Ray Clients currently support IPsec VPN concentrators from Cisco and Netscreen (Juniper).

For more information, see [Chapter 14, Sun Ray Client Firmware](#).

2.1.3. IP MultiPathing (Oracle Solaris 10)

Sun Ray Software supports arbitrary IP MultiPathing, or IPMP. IPMP provides failure detection and transparent network access failover for a system with multiple interfaces on the same IP link. IPMP can also provide load spreading of packets for systems with multiple interfaces.

This feature can be very useful on a Sun Ray server by increasing its network availability and performance. IPMP is supported only on Sun Ray servers running Oracle Solaris 10 in a shared network configuration (LAN with fully-routed subnets).

For more information about the IPMP feature in Oracle Solaris and how to configure it, see the [System Administration Guide: IP Services](#) manual.

When configuring IPMP, use the `if_mpadm` command to test NIC failure.

2.1.4. IPv4 and IPv6

Sun Ray Software supports both the IPv4 and IPv6 internet protocols. By default, the Sun Ray Clients are configured for the IPv4 protocol. For the Sun Ray Clients to work with IPv6, you need to update the firmware configuration on each client with by using the Configuration GUI or remote configuration file. See [Chapter 14, Sun Ray Client Firmware](#) for more information.

2.1.5. Network Performance Considerations

2.1.5.1. Packet Loss

The Sun Ray Software protocol is designed to operate well in conditions where other protocols would fail. However, if you detect sustained packet loss greater than 10 percent in the network, it may indicate other network problems. See [Chapter 20, Performance Tuning](#) for help.

2.1.5.2. Latency

Network latency between any Sun Ray client and its server is an important determinant of the quality of the user experience. The lower the latency, the better; latencies under 50 milliseconds for round trip delay are preferred. However, like familiar network protocols, the Sun Ray Client does tolerate higher latencies, but with degraded performance. Latencies up to 300 milliseconds provide usable, if somewhat sluggish, performance.

2.1.5.3. Out-of-Order Packets

Sun Ray Clients can tolerate small occurrences of out-of-order packet delivery, such as might be experienced on an Internet or wide-area intranet connection. Current Sun Ray firmware maintains a reordering queue that restores the correct order to packets when they are received out of order.

The process used to reorder packets can handle up to eight packets in a row that are out of order. For example, if the packets arrive as 7, 6, 5, 4, 3, 2, 1, the reorder process will hold packets 2 through 7 until packet 1 arrives and then it will process the packets in the correct sequence. Typically, packets do not get out of order unless they are traveling over a complex wide area network. Most out-of-order packets occur when packets can travel over a choice of paths, and most corporate networks provide very few redundant paths for packets to be routed over.

2.2. Configuring Sun Ray Server Discovery

There are two types of services that need to be discovered and used by a client:

- A firmware server that provides the latest firmware for a Sun Ray Client.
- A session server that provides a Sun Ray session to a client.

By default, these services are provided by the same server but they may be separated. There are many ways to configure how these servers are discovered. This section describes using pre-defined DNS entries to enable clients to find the servers. For other discovery methods that may provide more granularity or flexibility, see [Chapter 19, Alternate Network Configurations](#).

The on-screen display (OSD), when enabled, shows status during a client's server discoveries. For example, during DNS lookups, a status line in the OSD window shows the name being looked up and, if one is found, the IP address.

2.2.1. Firmware Server

Starting with Sun Ray Software 5.3, the firmware for Sun Ray Clients (called Sun Ray Operating Software) must be downloaded and installed separately on Sun Ray servers. Any Sun Ray server providing the latest Sun Ray Operating Software for Sun Ray Clients is considered a firmware server. For details, see [Section 3.2.4, "Installing Firmware Before Sun Ray Software Installation"](#).

When a Sun Ray Client boots in a properly configured environment, it checks with a firmware server to determine if it needs a Sun Ray Operating Software update. A Sun Ray Client's firmware server is discovered in the following order:

1. Locally configured value (configured through Configuration GUI)
2. DHCP Sun Ray vendor option (FWSrvr)
3. Standard DHCP option 66 (TftpSrvr) (IP Address or DNS name)
4. DNS lookup of `sunray-config-servers` (if mapped to multiple addresses, choose one randomly)

Each of these values are attempted in order until one succeeds. Although it is the last value attempted, the DNS lookup is the recommended firmware discovery configuration, as described below.

If the local configuration value is used and fails, none of the others are attempted. This prevents the overwriting of custom-configured firmware in a situation where the controlling firmware server happens to be temporarily unresponsive. See [Section 13.2.11, "Sun Ray Client Boot Process"](#) for more details on how a Sun Ray Client finds its firmware server.

Once a firmware server is discovered by a Sun Ray Client, the client retrieves a parameter file (`.parms`) via TFTP. This file is used by the client to determine if its currently installed Sun Ray Operating Software is older than the version on the firmware server. If so, the newer firmware is automatically downloaded and installed on the client.

In the event of an error in the firmware download, error messages through OSD display icons (if enabled) provide additional information that can be useful in diagnosing and correcting the problem. See [Chapter 16, Troubleshooting Icons](#) for details.



Note

By default, a client's firmware uses the configuration provided by the Sun Ray server's `.parms` file, which provides a centralized mechanism to administer firmware. However, you can enable the Configuration GUI on a client, which enables users to modify a Sun Ray Client's local configuration. See [Chapter 14, Sun Ray Client Firmware](#) for details.

2.2.2. Session Server

Once a client resolves whether it has the latest firmware installed, the next step in the boot process is to obtain a session from a Sun Ray server. As with discovering a firmware server, the client searches for a session server in the following order:

1. Locally configured value (configured through Configuration GUI)
2. Standard DHCP option 49 (IP Address or DNS name)
3. The `servers=` key in the client's `.parms` file
4. DNS lookup of `sunray-servers` (if mapped to multiple addresses, choose one randomly)

Each of these values are attempted in order until one succeeds. Although it is the last value attempted, the DNS lookup is the recommended session server discovery, as described in [Section 2.2.3, "Using Domain Name Service \(DNS\)"](#).

See [Section 13.2.11, "Sun Ray Client Boot Process"](#) for more details on how a Sun Ray Client finds its session server.

2.2.3. Using Domain Name Service (DNS)

Although there are multiple ways to configure server discovery, the recommended way is through DNS entries. If the Sun Ray DNS entries are defined appropriately for the Sun Ray Clients, no extra DHCP parameters are required by the Sun Ray Client beyond the basic network information. When the default DNS method for server resolution is used, the TFTP transport is the only method available for Sun Ray Client configuration and firmware updates.

The DNS entries for Sun Ray server discovery are as follows:

- `sunray-config-servers` for firmware servers
- `sunray-servers` for session servers

In both cases, if the DNS entry contains multiple server addresses, one is picked randomly. And, both entries should consist of several servers in your failover group for redundancy purposes.



Note

A DNS client incorporated in a Sun Ray Client's firmware allows many values to be names rather than IP addresses. Most values can be either a name or an IP address. If a name is specified, the DNS lookup appends the DHCP (or Configuration GUI) provided domain name. Components of the domain name are stripped successively until the lookup succeeds or only two components are left. If none of those lookups succeed, the name is looked up by itself. If the name itself ends with a dot character ("."), the name is taken to be a rooted name, and it is looked up without domain name components appended.

Chapter 3. Installing and Configuring

Table of Contents

3.1. Product Requirements	13
3.1.1. Operating System Requirements	14
3.1.2. Sun Ray Operating Software	14
3.1.3. Windows Remote Desktop Support	14
3.1.4. Feature Differences Between Oracle Solaris and Oracle Linux Platforms	15
3.1.5. Differences Between Oracle Solaris 10 and Oracle Solaris 11 Platforms	16
3.1.6. Disk Space Requirements	16
3.1.7. Oracle Solaris 10 Prerequisites	16
3.1.8. Oracle Solaris 11 Prerequisites	17
3.1.9. Oracle Linux Prerequisites	18
3.1.10. Java Runtime Environment (JRE) Requirements	21
3.1.11. Sun Ray Admin GUI Web Server Requirements	21
3.1.12. Sun Ray Admin GUI Web Browser Requirements	22
3.1.13. Sun Ray Data Store Port Requirements	22
3.1.14. Ports and Protocols	23
3.2. Installing	26
3.2.1. Using the <code>utsetup</code> Command	26
3.2.2. Not Using the <code>utsetup</code> Command	26
3.2.3. Automating Sun Ray Software Installations	27
3.2.4. Installing Firmware Before Sun Ray Software Installation	27
3.2.5. How to Install Sun Ray Software	28
3.2.6. Post-Installation Configuration	30
3.2.7. How to Install the Windows Connector Components on a Windows System	31
3.2.8. How to Clone a Sun Ray Server	34
3.2.9. How to Install and Configure a Sun Ray Server With Default Settings	36
3.2.10. How to List the Current Sun Ray Software Version	37
3.2.11. How to Remove Sun Ray Software	37
3.2.12. Installation (<code>utinstall</code>) Error Messages	37
3.3. Configuring Oracle Solaris 11 Trusted Extensions	39
3.3.1. How to Configure Sun Ray Software on Oracle Solaris 11 Trusted Extensions	39
3.4. Configuring Oracle Solaris 10 Trusted Extensions	41
3.4.1. How to Configure a Private Network on Oracle Solaris 10 Trusted Extensions	41
3.4.2. How to Configure Shared Multilevel Ports (MLP) for Sun Ray Services	42
3.4.3. How to Increase the Number of X Server Ports	43
3.4.4. How to Configure the Windows Connector on Oracle Solaris Trusted Extensions	43
3.5. Upgrading	45
3.5.1. Installing Firmware Before Sun Ray Software Upgrade	45
3.5.2. How to Upgrade Sun Ray Software	45
3.5.3. Planning Upgrades Using Failover Groups	49
3.5.4. How to Preserve Sun Ray Software Configuration Data	50

This chapter provides an overview of installing and configuring Sun Ray Software on a system, which makes it a Sun Ray server.

3.1. Product Requirements

This section provides the product requirements for the Sun Ray Software 5.4 release.

3.1.1. Operating System Requirements

[Table 3.1, “Supported Sun Ray Software Operating Systems”](#) lists the supported Sun Ray Software operating systems for the Sun Ray Software 5.4 release.

Table 3.1. Supported Sun Ray Software Operating Systems

Operating System	Supported Releases
Oracle Solaris 10 on SPARC and x86 platforms	<ul style="list-style-type: none">• Oracle Solaris 10 8/11 or later• Oracle Solaris 10 8/11 or later with Trusted Extensions
Oracle Solaris 11 on SPARC and x86 platforms	<ul style="list-style-type: none">• Oracle Solaris 11.1 or later• Oracle Solaris 11.1 or later with Trusted Extensions
Oracle Linux on x86 platform (64-bit)	<ul style="list-style-type: none">• Oracle Linux 5.8• Oracle Linux 6.3

Sun Ray Software is supported and can be installed in an Oracle virtualized environment. If you encounter a problem when using an unsupported virtualization environment, you may be asked to demonstrate the issue on a non-virtualized operating system to ensure the problem is not related to the virtualization product.

Both Oracle Solaris and Oracle Solaris with Trusted Extensions can use zones to permit multiple virtualized operating system environments to coexist in a single instance of Oracle Solaris, allowing processes to run in isolation from other activity on the system for added security and control. Sun Ray Software is supported only in the global zone.



Note

Oracle products certified on Oracle Linux are also certified and supported on Red Hat Enterprise Linux due to implicit compatibility between both distributions. Oracle does not run any additional testing on Red Hat Enterprise Linux products.

To prepare the servers before installing the Sun Ray Software, see [Section 3.1.7, “Oracle Solaris 10 Prerequisites”](#), [Section 3.1.8, “Oracle Solaris 11 Prerequisites”](#), and [Section 3.1.9, “Oracle Linux Prerequisites”](#).

3.1.2. Sun Ray Operating Software

The Sun Ray Operating Software 11.1.1 for Sun Ray Clients is required for many of the new Sun Ray Software 5.4 features. Sun Ray Operating Software is the official name of the Sun Ray Client firmware.

See [Section 3.2.4, “Installing Firmware Before Sun Ray Software Installation”](#) for details on how to install the Sun Ray Operating Software on Sun Ray Clients.

3.1.3. Windows Remote Desktop Support

The following Windows remote desktops are supported with Sun Ray Software:

- **Windows XP Professional with SP2 (64-bit)**
- **Windows XP Professional with SP3 (32-bit)**
- **Windows Server 2003 R2 SP2 Enterprise Edition (32-bit and 64-bit)**
- **Windows 7 SP1 Enterprise Edition (32-bit and 64-bit)**

Sun Ray Software is tested on and supports Windows 7 SP1 Enterprise Edition. Windows 7 Professional, Ultimate, or other Enterprise editions can be used, but in the event of an issue, you must be able to reproduce the issue in Windows 7 SP1 Enterprise Edition.

Multiple monitor and audio recording (input audio) are only available in Windows 7 Ultimate and Enterprise editions. Windows 7 Professional does support a single desktop spanned across multiple monitors (spanned mode).

- **Windows Server 2008 R2 SP1 Enterprise Edition (64-bit)**
- **Windows 8 Enterprise Edition (32-bit and 64-bit)**

Sun Ray Software is tested on and supports Windows 8 Enterprise Edition. Windows 8 Professional can be used, but in the event of an issue, you must be able to reproduce the issue in Windows 8 Enterprise Edition.

- **Windows Server 2012 (64-bit)**

Table 3.2, “Supported Features for Windows Remote Desktops” shows what features are supported for each Windows remote desktop. Some Windows releases require a Windows connector component to be installed for specific feature support. For detailed information, see [Section 3.2.7, “How to Install the Windows Connector Components on a Windows System”](#).

Table 3.2. Supported Features for Windows Remote Desktops

	Windows XP SP2 (64-bit)	Windows XP SP3 (32-bit)	Windows Server 2003 R2 SP2 (32-bit/64-bit)	Windows 7 SP1 (32-bit/64-bit)	Windows Server 2008 R2 SP1 (64-bit)	Windows 8 (32-bit/64-bit)	Windows Server 2012 (64-bit)
Video Acceleration	✓	✓	✓	✓	✓		
USB Redirection	✓	✓	✓	✓	✓	✓	✓
Audio Input	✓	✓	✓	✓	✓	✓	✓
Enhanced Network Security	✓	✓	✓	✓	✓	✓	✓
Session Directory/Session Broker	n/a	n/a	✓	n/a	✓	n/a	✓
Smart Card Services	✓	✓	✓	✓	✓		



Note

Video acceleration support is dependent on the Windows desktop version, the application being used, and the client used to connect to the desktop. See [Section 17.5, “Video Acceleration”](#) for details.

3.1.4. Feature Differences Between Oracle Solaris and Oracle Linux Platforms

The following Sun Ray Software features are not supported on a Sun Ray server running the Oracle Linux platform.

- Using mass storage devices without the USB redirection Windows component provides much lower performance on Oracle Linux than Oracle Solaris due to the design of the Oracle Linux mass storage subsystem. Use USB redirection for optimum performance with mass storage devices.

- Predefined kiosk session types are not available, which provide a desktop, a window manager, and the ability to configure a set of applications. Sun Java Desktop (JDS), Release 3, is an example of a predefined session type provided for Oracle Solaris 10. See [Section 10.1, “Kiosk Overview”](#) for more information.
- The CCID IFD handler, which provides access to external CCID-compliant USB smart card readers connected to desktop clients, is not supported on Sun Ray servers running Oracle Linux.
- The scbus v1 smart card protocol is not supported on Sun Ray servers running Oracle Linux.
- When hotdesking without smart cards, the Options menu in the NSCM login screen is not available for Oracle Linux. This includes both the QuickLogin and Exit options. See [Section 9.2.3, “How to Log in to an NSCM Session”](#) for details.

3.1.5. Differences Between Oracle Solaris 10 and Oracle Solaris 11 Platforms

The following list describes the differences between a Sun Ray server running the Oracle Solaris 10 and Oracle Solaris 11 platforms.

- Predefined kiosk session types are available only for Oracle Solaris 10. There are no predefined kiosk session types for Oracle Solaris 11.
- IP MultiPathing (IPMP) is supported only on Sun Ray servers running Oracle Solaris 10 in a shared network configuration (LAN with fully-routed subnets).

3.1.6. Disk Space Requirements

[Table 3.3, “Disk Space Requirements”](#) lists the disk space requirements for specific directories.

Table 3.3. Disk Space Requirements

Default Installation Path	Requirements
/	1 Mbyte
/etc/opt/SUNWut/srds	0.1 Mbytes
/opt	70 Mbytes
/opt/SUNWut/srds	4.6 Mbytes
/var/adm/log (Oracle Solaris)	5 Mbytes
/var/log (Oracle Linux)	2.5 Mbytes
/var/opt/SUNWut	Allow enough disk space for the data store and log files. For 1,000 entries, allocate roughly 1.5 Mbytes of disk space, 64 Mbytes of RAM, and 528 Mbytes of swap space.
/var/tmp	5 Mbytes

3.1.7. Oracle Solaris 10 Prerequisites

This section describes the prerequisites when using Oracle Solaris 10 for a Sun Ray server:

- The Entire Distribution software cluster is required and must be installed.
- The latest Recommended Patchset must be installed prior to the Sun Ray Software installation, which you can download from [My Oracle Support](#).
- The Common Desktop Environment (CDE) might not be available in a future Oracle Solaris 10 release. Users should migrate to the Java Desktop System. CDE will not be supported on future versions of Sun Ray Software when CDE is officially removed from the Oracle Solaris 10 release.

- To increase the performance of the Sun Ray Clients, make the following configuration update:

1. Add the following line to the Sun Ray server's `/etc/system` file.

```
set hires_tick=1
```

For more information about this setting, see [Section 20.3, “How to Improve Sun Ray Client Performance by Decreasing Buffering on the Network Switch \(Oracle Solaris\)”](#).

2. Reboot the Sun Ray server.

3.1.8. Oracle Solaris 11 Prerequisites

This section describes the prerequisites when using Oracle Solaris 11 for a Sun Ray server:

- The default Oracle Solaris 11 packages, which are provided through the `solaris` package publisher, are required for Sun Ray Software and must be installed on the Sun Ray server.
- The latest Oracle Solaris 11 Support Repository Update must be installed on the Sun Ray server.
- Additional packages are required, which you can install using the `utpkgcheck` command provided in the Sun Ray Software media pack.

The `utpkgcheck` command uses the Oracle Solaris 11 Image Packaging System (IPS) to install the additional packages. The `utpkgcheck` command uses the repository URI that is configured for the `solaris` package publisher.

Use the following command to install the additional packages on an Oracle Solaris 11 server:

```
# utpkgcheck -i
```

- To increase the performance of the Sun Ray Clients, make the following configuration updates:

1. Add the following line to the Sun Ray server's `/etc/system` file.

```
set hires_tick=1
```

For more information about this setting, see [Section 20.3, “How to Improve Sun Ray Client Performance by Decreasing Buffering on the Network Switch \(Oracle Solaris\)”](#).

2. Add the following line to the Sun Ray server's `/etc/system` file.

```
set mac:mac_cpu_binding_on=0
```

For more information about this setting, see [Section 20.2, “How to Improve Network Performance by Disabling CPU Binding \(Oracle Solaris 11\)”](#).

3. Reboot the Sun Ray server.

- To provide desktop client users an optimized desktop, enable the multi-user desktop service on the Sun Ray server:

```
# svcadm enable application/gconf/multi-user-desktop
```

See [Optimizing the Oracle Solaris 11 Desktop for a Multi-User Environment](#) for more details.

- To optimize the shared memory used by PulseAudio, add the following line to the `/etc/pulse/client.conf` file on the Sun Ray server:

```
shm-size-bytes = 131072
```

See [Section 13.1.8, “Audio Output Troubleshooting \(Oracle Solaris 11 and Oracle Linux 6\)”](#) for details about PulseAudio.

3.1.9. Oracle Linux Prerequisites

This section describes the prerequisites when using Oracle Linux for a Sun Ray server:

- For Oracle Linux 5, the default package set is required for Sun Ray Software and must be installed on the Sun Ray server.
- For Oracle Linux 6, the [Desktop](#) package set is required for Sun Ray Software and must be installed on the Sun Ray server.
- Additional packages are required, which you can install using the [utpkgcheck](#) command provided in the Sun Ray Software media pack. See [Section 3.1.9.1, “How to Install Required Packages Using utpkgcheck”](#) for details.
- The firewall and SELinux services must be disabled.

For Oracle Linux 6, you must disable these services after installing the operating system. To disable firewall services, use the Firewall Configuration dialog (System > Administration > Firewall). To disable SELinux services, edit the `/etc/selinux/config` file as follows and restart the server:

```
SELINUX=disabled
```

- For Oracle Linux 6, optimize the shared memory used by PulseAudio. Add the following line to the `/etc/pulse/client.conf` file on the Sun Ray server:

```
shm-size-bytes = 131072
```

See [Section 13.1.8, “Audio Output Troubleshooting \(Oracle Solaris 11 and Oracle Linux 6\)”](#) for details about PulseAudio.

3.1.9.1. How to Install Required Packages Using [utpkgcheck](#)

The [utpkgcheck](#) command uses the [yum](#) command to retrieve and install the required packages for Sun Ray Software, which relies on the server being configured with Oracle Unbreakable Linux Network (ULN) or the Oracle Public Yum Server. For servers running Oracle 6, you must also configure yum to include the multiseat GDM repository, which is an enhanced version of GDM that supports multi-seat capability required in Sun Ray environments.

If the server does not have the ULN or Public Yum Server configured, [utpkgcheck](#) will try to install the necessary packages from the automounted Oracle Linux DVD if available. This option is useful only for servers running Oracle Linux 5, because the multiseat GDM channel is not required.

Use the following steps to install the required packages:

1. Make sure yum is configured properly on the server, as described in [Section 3.1.9.2, “Configuring ULN Channels for utpkgcheck”](#) and [Section 3.1.9.3, “Configuring Public Yum Repositories for utpkgcheck”](#).



Note

Make sure to unsubscribe from the specified *_latest ULN channels or disable the specified *_latest yum repositories as instructed. Otherwise, the server will update to a later version of Oracle Linux that is not supported by this version

of Sun Ray Software. If there are any Oracle Linux bugs affecting Sun Ray Software that have been fixed through an Oracle Linux patch, you must use ULN to get those fixes.

2. Install the required packages on the Oracle Linux server:

```
# utpkgcheck -i
```

In some instances, you must reboot the system if the following warning message is displayed:

```
WARNING: System must be rebooted in order to complete installation.
```

3. Update the server with the latest package versions.

```
# yum update
```

4. Reboot the system.

```
# reboot
```

5. (Oracle Linux 6 only). To enable root login to GDM, comment out or remove the following line from the `/etc/pam.d/gdm` file.

```
auth required pam_succeed_if.so user != root quiet
```

The multiseat GDM repository disables root login by default.

3.1.9.2. Configuring ULN Channels for `utpkgcheck`

Before running `utpkgcheck`, make sure the server is subscribed to the required channels on ULN. For information about ULN, see the [Oracle Linux Unbreakable Linux Network User's Guide](#).

After you have enabled and disabled repositories, use the `yum clean all` command to clear the yum cache, and then use the `yum repolist` command to check that you have enabled the correct repositories.

ULN Channel Subscriptions for Oracle Linux 6 Platforms

- Unsubscribe from the following channels:

Channel Label	Channel Name
ol6_x86_64_latest	Oracle Linux 6 Latest (x86_64)
ol6_x86_64_UEK_latest	Latest Unbreakable Enterprise Kernel for Oracle Linux 6 (x86_64)

- Subscribe to the following channels:

Channel Label	Channel Name
ol6_u3_x86_64_patch	Oracle Linux 6 Update 3 Patch (x86_64)
ol6_u3_x86_64_base	Oracle Linux 6 Update 3 installation media copy (x86_64)
ol6_x86_64_UEK_base	Unbreakable Enterprise Kernel for Oracle Linux 6 (x86_64)
ol6_x86_64_gdm_multiseat	Oracle Linux 6 GDM Multiseat

ULN Channel Subscriptions for Oracle Linux 5 Platforms

- Unsubscribe from the following channels:

Channel Label	Channel Name
el5_x86_64_latest	Enterprise Linux 5 Latest (x86_64)
ol5_x86_64_latest	Oracle Linux 5 Latest (x86_64)
ol5_x86_64_UEK_latest	Latest Unbreakable Enterprise Kernel for Oracle Linux 5 (x86_64)

- Subscribe to the following channels:

Channel Label	Channel Name
ol5_u8_x86_64_patch	Oracle Linux 5 Update 8 Patch (x86_64)
ol5_u8_x86_64_base	Oracle Linux 5 Update 8 installation media copy (x86_64)
ol5_x86_64_UEK_base	Unbreakable Enterprise Kernel for Oracle Linux 5 (x86_64)

3.1.9.3. Configuring Public Yum Repositories for `utpkgcheck`

Before running `utpkgcheck`, make sure the server has the appropriate repositories configured for the Public Yum Server. For information about the Public Yum Server, see <http://public-yum.oracle.com>. For Oracle Linux 6, see the [Oracle Linux Administrator's Solutions Guide for Release 6](#).



Note

For Oracle Linux 6 platforms, you must download the latest yum configuration file (<http://public-yum.oracle.com/public-yum-ol6.repo>) and copy it to the `/etc/yum.repos.d` directory on the server. The latest yum configuration file contains entries for the required Oracle Linux 6 GDM Multiseat repository.

After you have enabled and disabled repositories, use the `yum clean all` command to clear the yum cache, and then use the `yum repolist` command to check that you have enabled the correct repositories.

Repository Configuration for Oracle Linux 6 Platforms

- Disable the following repositories:

Repository	Name
<code>[ol6_latest]</code>	Oracle Linux 6 Latest (x86_64)
<code>[ol6_UEK_latest]</code>	Latest Unbreakable Enterprise Kernel for Oracle Linux 6 (x86_64)

- Enable the following repositories:

Repository	Name
<code>[ol6_u3_base]</code>	Oracle Linux 6 Update 3 installation media copy (x86_64)
<code>[ol6_UEK_base]</code>	Unbreakable Enterprise Kernel for Oracle Linux 6 (x86_64)
<code>[ol6_gdm_multiseat]</code>	Oracle Linux 6 GDM Multiseat (x86_64)

Repository Configuration for Oracle Linux 5 Platforms

- Disable the following repositories:

Repository	Name
<code>[ol5_latest]</code>	Oracle Linux 5 Latest (x86_64)

Repository	Name
<code>[ol5_UEK_latest]</code>	Latest Unbreakable Enterprise Kernel for Oracle Linux 5 (x86_64)

- Enable the following repositories:

Repository	Name
<code>[ol5_u8_base]</code>	Oracle Linux 5 Update 8 installation media copy (x86_64)
<code>[ol5_UEK_base]</code>	Unbreakable Enterprise Kernel for Oracle Linux 5 (x86_64)

3.1.10. Java Runtime Environment (JRE) Requirements

Sun Ray Software 5.4 Admin GUI requires a 32-bit implementation of a Java(TM) 2 Platform, Standard Edition JRE(TM) of at least 1.6. To check what JRE version is installed on your system, use the following command:

```
# java -version
```

It is recommended that you install the latest Java release, which is available at <http://www.oracle.com/technetwork/java/javase/downloads>. A supported version of JRE is bundled in the unzipped Sun Ray Software media pack in the [Supplemental](#) directory.

The Sun Ray Software installation script assumes JRE is installed in the `/usr/java` directory by default. For example, if you want to accept the default when installing the Sun Ray Software on an Oracle Linux server, install JRE 1.6 or later on the server and then create a symlink from `/usr/java` to the newly created `jre` directory. The following sequence of commands installs the JRE in the `/usr` directory and creates a symbolic link from `/usr/java` to the new `jre1.6.0_version` directory.

```
# cd /usr
# Supplemental-dir/Java_Runtime_Environment/Linux/jre-6uversion-linux-i586.bin
# ln -s jre1.6.0_version /usr/java
```



Note

A 64-bit JRE is not suitable for use with Sun Ray Software. The 32-bit JRE is required, even when the platform is capable of supporting a 64-bit JRE.



Note

If you are using JRE version 1.6 on a server running Oracle Solaris 11, a [Secure Connection Failed](#) error may occur when launching the Admin GUI through a secure URL. To resolve this problem, either update the JRE version to 1.7 or disable TLS 1.0 encryption in the browser's preferences.

3.1.11. Sun Ray Admin GUI Web Server Requirements

The Sun Ray Administration Tool (Admin GUI) requires that a Web server be installed and running on each Sun Ray server. The Admin GUI must be hosted in a web container that supports the JavaServlet 2.4 and JavaServer Pages 2.0 specification. The Apache Tomcat 5.5 Web container implements these standards and runs on any operating system that has a Java Runtime Environment (JRE).

The `utconfig` script prompts for the location of an Apache Tomcat HTTP Server and asks whether it should be configured automatically.

- To configure the server automatically, supply the path and answer **Yes**.
- To configure the HTTP server later by using the `utconfig -w` command, answer **No**.

The Sun Ray configuration script uses port 1660 for the Sun Ray Administration Tool (Admin GUI) by default. If this port is unavailable, you can configure a new port while running the `utconfig` command.

An Apache Tomcat 5.5 archive is included in the Sun Ray Software media pack under [Supplemental/Apache_Tomcat](#). The most recent version of Tomcat 5.5 can be downloaded from <http://tomcat.apache.org>.

See [How to Install Apache Tomcat](#) for details.

3.1.11.1. How to Install Apache Tomcat

If Tomcat 5.5 is already installed on your system, you can omit the steps below and specify the path, if necessary, during the Sun Ray Software configuration.

1. As superuser, open a shell window on the Sun Ray server.

```
% su -
```

2. Change to the [Apache_Tomcat](#) directory. For example:

```
# cd media_pack_dir/Supplemental/Apache_Tomcat
```

3. Extract the Tomcat archive into a suitable directory, such as `/opt`.

For Oracle Solaris

The Tomcat archive uses GNU tar extensions and must be untarred with a GNU-compatible version of the `tar` command, such as `gtar`.

```
# /usr/sfw/bin/gtar -xvz -C /opt -f apache-tomcat-5.5.36.tar.gz
```

For Oracle Linux

```
# tar -xvz -C /opt -f apache-tomcat-5.5.36.tar.gz
```

(Optional) Create a symbolic link to the default location for the Sun Ray Software installation script.

```
# ln -s apache-tomcat-5.5.36 /opt/apache-tomcat
```

3.1.12. Sun Ray Admin GUI Web Browser Requirements

The Sun Ray Administration Tool (Admin GUI) has been tested and works with the default browsers provided by the operating systems listed in [Table 3.1, "Supported Sun Ray Software Operating Systems"](#).

3.1.13. Sun Ray Data Store Port Requirements

When you configure a new Sun Ray server in a failover environment that uses Sun Ray Software 5.4 only, service port 7012 is used by default.

If you already have an LDAP (Lightweight Data Access Protocol) server configured on the Sun Ray server, it can coexist with the Sun Ray data store. However, it must not use port 7012, which is reserved for use by the Sun Ray data store.

If you configure a new Sun Ray server in a failover group with mixed versions of Sun Ray Software, you must make sure that the primary server is running Sun Ray Software 5.4.

If the secondary server is running Sun Ray Software 5.4, no special care is required. The `utreplica` utility automatically synchronizes with the port number on the primary.

**Note**

Although configuring mixed failover groups consisting of servers running various versions of Sun Ray Server Software is possible, this practice is discouraged. For more information, see [Chapter 6, Failover Groups](#).

3.1.14. Ports and Protocols

The following section summarize Sun Ray system port and protocol usage.

The range of dynamic/UDP and dynamic/TCP ports on the server is constrained to the range defined by the `utservices-low` and `utservices-high` UDP service definitions, whose default values in `/etc/services` are 40000 and 42000 respectively. This range should not be redefined to constrain ports too tightly. The range of ports must be sufficient to provide several ports per connected Sun Ray Client.

Ranges used by the client include the following:

- Dynamic/TCP ports on the client are in the range 32768-65535.
- Dynamic/UDP ports on the client are in the range 4096-65535.
- ALP rendering traffic (ALP-RENDER) always uses a UDP port number greater than 32767 at the client.

3.1.14.1. Sun Ray Client-to-Server Ports and Protocols

[Table 3.4, "Sun Ray Client-to-Server Ports and Protocols"](#) lists the Sun Ray Client-to-server ports and protocols. In the table, a double-headed arrow in the Flow column indicates the direction of the initial packet. In most cases, the client (a Sun Ray Client or Oracle Virtual Desktop Client) initiates the interaction.

Table 3.4. Sun Ray Client-to-Server Ports and Protocols

Client Port / Flow	Protocol	Server Port / Flow	Peer	Importance/ Comments
66/UDP (BOOTPC/ DHCP)	DHCP	67/UDP (BOOTPS/ DHCP)	DHCP Service	Mandatory.
broadcast=>>		<=broadcast		Network and configuration parameter discovery.
unicast=>>		<=unicast		
Dynamic/UDP	TFTP	69/UDP (TFTP)	TFTP Service	Recommended.
unicast=>>		<=unicast		Firmware download (Configuration parameter download).
Dynamic/UDP	DNS	53/UDP (domain)	DNS Service	Optional.
unicast=>>		<=unicast		For server name lookups.
514/UDP (syslog)	Syslog	514/UDP (syslog)	Syslog Service	Optional.
unicast=>>				Event reporting.
Dynamic/TCP	pcscd	4120/TCP (pcscd)	Sun Ray Server	Mandatory.
unicast=>>		<=unicast		PC/SC-lite smart card service.

Client Port / Flow	Protocol	Server Port / Flow	Peer	Importance/ Comments
				Used to be TCP port 5999.
Dynamic/UDP broadcast=>>>	ALP-DISCOVERY	7009/UDP (utauthd-gm) <=unicast	Sun Ray Server	Optional. On-subnet Sun Ray server discovery.
Dynamic/TCP unicast=>>	ALP-AUTH	7009/TCP (utauthd) <=unicast	Sun Ray Server	Mandatory. Presence, control, status.
Dynamic/UDP with port number >= 32768 unicast=> or unicast=>> when NAT is in use	ALP-RENDER	Dynamic/UDP constrained by utservices-low and utservices-high <<=unicast or <=unicast when NAT is in use	Sun Ray Server	Mandatory. On-screen drawing, user input, audio.
5498/UDP unicast=>>>	ALP-AUDIO-IN	Dynamic/UDP constrained by utservices-low and utservices-high	Sun Ray Server	Optional. Inbound audio.
Dynamic/TCP unicast=>>>	ALP-DEVMGR	7011/TCP (utdevmgr) <=unicast	Sun Ray Server	Optional. Device management.
Dynamic/TCP with port number >= 32768 unicast=>	ALP-DEVDATA	Dynamic/TCP constrained by utservices-low and utservices-high <<=unicast	Sun Ray Server	Optional. Device communication when accessing external device, including USB redirection.
7777/TCP unicast=>	ALP-DEVDATA	Dynamic/TCP <<=unicast	Sun Ray Server	Optional. Device communication when accessing external devices connected to Sun Ray Clients running older firmware. Not used with USB redirection.
7013/UDP (utquery) unicast=>	ALP-QUERY	Dynamic/UDP <<=unicast <<=broadcast	Any	Optional. utquery support.

**Note**

Due to CR 12301209, the keyboard may become unresponsive to input. To work around this issue, allow ICMP messages to flow from the Sun Ray server to the client.

3.1.14.2. Sun Ray Server-to-Server Protocols

Table 3.5, “Sun Ray Server-to-Server Ports” lists the Sun Ray server-to-server ports. In the table, a double-headed arrow indicates the direction of the initial packet.

Table 3.5. Sun Ray Server-to-Server Ports

Sun Ray Server Port	Protocol	Port	Peer	Notes
	<=>ARP=>		All on subnet	IP-to-MAC mapping.
	<=>ICMP ECHO=>		Any	Admin: presence.
Transient	SYSLOG/UDP unicast=>	514 (SYSLOG)	Syslog Server	Status reporting, if required.
1660 (HTTP)	<=>HTTP/TCP=>	Transient	Localhost	Admin GUI, if configured.
1661 (HTTPS)	<=>HTTPS/TCP=>	Transient	Localhost	Admin GUI, if configured.
7007 (UTSESSIOND)	<=>UTSESSION/TCP=>	Transient	Any	Session members.
7007 (UTSESSIOND)	<=>UTSESSION/TCP=>	Privileged	Localhost	Session management.
7008 (UTRCMD)	<=>UTRCMD/TCP=>	Privileged	Sun Ray Group Member	Remote execution.
7009 (UTAUTHD)	<=>UTAUTHD-GM/UDP=> broadcast or multicast	7009 (UTAUTHD)	Sun Ray Server	Group discovery, if required.
7010 (UTAUTH-CB)	<=>UTAUTH-CB/TCP=>	Transient	Any	Admin: control and status.
7011 (UTDEVMGRD)	<=>UTDEVMGRD/TCP=>	7011 (UTDEVMGR)	Sun Ray Group Member	Device control and status.
7011 (UTDEVMGR)	<=>UTDEVMGR/TCP=>	Transient	Any	Device clients.
7012 (UTDS)	<=>UTDS/TCP=>	Transient	Any	Data store, if required.
7014 (UTTSCPD)	<=>UTTSCPD/TCP=>	Privileged	Sun Ray Windows Connector from Localhost	Bridge/Proxy between Windows connector and the Sun Ray server.

3.1.14.3. Windows Connector

For basic Windows connector operations (RDP port access), the Windows server firewall needs TCP port 3389 open for inbound connections. The Sun Ray server (where the Windows connector is running) firewall needs TCP port 3389 open for outbound connections.

3.1.14.4. Multimedia Redirection

For multimedia redirection on Windows XP and Windows Server 2003 R2, the Windows server firewall must have a TCP port between 6000 and 10000 open for inbound connections. The Sun Ray server (where the Windows connector is running) firewall must have a TCP port between 6000 and 10000 open for outbound connections.

3.2. Installing

This section provides detailed information about how to install Sun Ray Software.

3.2.1. Using the `utsetup` Command

The `utinstall` and `utconfig` commands are the basic commands to install and configure Sun Ray software on a system. There are also a number of additional commands that you need to run to configure a basic Sun Ray server and make it ready to provide sessions to clients.

The `utsetup` command provides a way to run all the appropriate commands, including `utinstall` and `utconfig`, in the appropriate sequence to install and configure a Sun Ray server. By design, the `utsetup` command configures a Sun Ray server to use a shared network (LAN) and it generates a set of `.parms` files in the TFTP home directory for managing Sun Ray Client firmware. This recommended configuration is detailed in [Chapter 2, *Planning a Sun Ray Network Environment*](#).

When you use `utsetup` command to initially install a Sun Ray server, it runs the following commands in sequence:

- `utinstall`
- `utconfig`
- `utpolicy -a -z both -g -M`
- `utreplica` (if HA/failover group selected)
- `utfwadm -A -a -V` (optional)
- `utadm -L on` (optional)
- `utstart -c`

See [Section 3.2.5, “How to Install Sun Ray Software”](#) for installation instructions using the `utsetup` command.

3.2.2. Not Using the `utsetup` Command

If you need to install and configure a Sun Ray server in a different way than what the `utsetup` command provides, you must run the individual commands as noted in [Section 3.2.1, “Using the `utsetup` Command”](#). The following situations may force you to do this:

- Using Kickstart (Oracle Linux), Automated Installer (Oracle Solaris 11), or JumpStart (Oracle Solaris 10) to install the software.
- Configuring the Sun Ray Software without the Sun Ray data store, which is known as zero administration mode.

This document does not provide detailed instructions about how to use the individual commands instead of using the `utsetup` command. Refer to the man pages for detailed information about the alternative commands.

3.2.3. Automating Sun Ray Software Installations

The `utsetup` command enables you to clone the Sun Ray Software installation and configuration process by recording the user responses to the installation prompts and then using those responses at another time and even on another Sun Ray server. User responses are stored in the `/var/opt/SUNWut/utdialog.d/*.utdialog_responses.props` files and are known as response files.

With `utsetup` and response files, you have the ability to clone a Sun Ray server installation and configuration setup or to provide default settings for a hands-free, automated installation and configuration solution. Cloning Sun Ray servers can be helpful in many situations, including setting up a number of Sun Ray servers in a failover group.



Note

Using the `utsetup` command without any options performs an actual install and configuration on the server you run it on.

The following commands can use the information recorded in the response files:

- `utsetup`
- `utinstall`
- `utconfig`
- `utpolicy`
- `utpw`
- `utgroupsig`
- `utreplica`

See the following procedures for automating Sun Ray Software installations:

- [Section 3.2.8, “How to Clone a Sun Ray Server”](#)
- [Section 3.2.9, “How to Install and Configure a Sun Ray Server With Default Settings”](#)

3.2.4. Installing Firmware Before Sun Ray Software Installation

Starting with Sun Ray Software release 5.3, the Sun Ray Operating Software, formerly known as Sun Ray Client firmware, is no longer included with Sun Ray Software and must be downloaded from [My Oracle Support](#) separately. Updating the Sun Ray Clients with the latest Sun Ray Operating Software ensures that the latest Sun Ray Software features are provided. The Sun Ray Client firmware is now officially called Sun Ray Operating Software, but the term “firmware” will continue to be used throughout the documentation. See [Sun Ray Operating Software 11.1.1 Release](#) for details.

If you download and install the latest Sun Ray Operating Software before you install Sun Ray Software on a server, the `utsetup` command enables you to configure the installed firmware and make it available to clients through the `utfwadm` command similar to previous releases. The current installation procedure provides steps on how to download and install the firmware.

Installing the latest Sun Ray Operating Software before installing Sun Ray Software is recommended, but you can still install and configure the firmware after installing Sun Ray Software. To update the firmware on Sun Ray Clients outside of the Sun Ray Software installation process, refer to [Section 14.3, “How to Update Firmware on Sun Ray Clients”](#).

If the firmware is not installed on the server, the Sun Ray Software installation script will provide the following warning:

```
Sun Ray Operating Software (firmware) is not installed.
It is recommended that you install the latest firmware
before installing Sun Ray Software.
Continue without firmware? (Y/N) [Y]
```

Also, if an older firmware version is installed on the server, the Sun Ray Software installation script will provide the following warning:

```
You are attempting to install SRS on a system with an old version
of Sun Ray Operating Software.
Continue with old firmware? (Y/N) [N]
```

3.2.5. How to Install Sun Ray Software

This procedure uses the `utsetup` command to install and configure a Sun Ray server for both the Oracle Linux or Oracle Solaris operating system.

The `utsetup` command optionally configures a Sun Ray server to use a shared network (LAN) and generates a set of `.parms` files in the TFTP home directory for managing Sun Ray Client firmware (DHCP is not configured to manage firmware downloads). This configuration is detailed in [Chapter 2, *Planning a Sun Ray Network Environment*](#).

Before You Begin

- Make sure the server targeted to become the Sun Ray server meets the [Sun Ray Software product requirements](#), including the prerequisites sections.
- A Sun Ray server requires both a fixed host name and a static IP address. A Sun Ray server cannot be a DHCP client.
- It is recommended that you download and install the latest Sun Ray Operating Software (firmware) before starting the Sun Ray Software installation. Updating the Sun Ray Clients with the latest firmware ensures that the latest Sun Ray Software features are provided.
- (Oracle Linux only) The Sun Ray Software installation script removes the Shutdown/Restart options from the console; however, you can open a terminal and execute the commands.
- (Oracle Linux only) When Sun Ray Software is activated as part of the installation process, the NetworkManager service on the Sun Ray server is disabled to avoid DHCP and network service conflicts. If there are scenarios where the NetworkManager service is needed (using a mobile Sun Ray server for demo purposes), then you must manually configure the server's NICs. When you uninstall Sun Ray Software, the NetworkManager service is enabled to restore the default configuration.

Steps

1. Download and unzip the Sun Ray Software 5.4 media pack and make it accessible to the Sun Ray server.

See <http://www.oracle.com/technetwork/server-storage/sunrayproducts/downloads/index.html>

2. (Optional) Download and unzip the latest Sun Ray Operating Software (firmware) and make it accessible to the Sun Ray server.

See <http://www.oracle.com/technetwork/server-storage/sunrayproducts/downloads/index.html>

Updating the Sun Ray Clients with the latest firmware ensures that the latest Sun Ray Software features are provided.

3. Become superuser on the Sun Ray server.

To avoid installation script errors that can occur if user environment settings are carried forward, use the following command:

```
% su - root
```

4. If you downloaded the latest Sun Ray Operating Software (firmware), change directory to the unzipped firmware directory and install the firmware to make it available to the Sun Ray Software installation script.

```
# ./utfwinstall
```

5. Change directory to the unzipped Sun Ray Software media pack and install Sun Ray Software on the Sun Ray server.

```
# ./utsetup
```

When the installation script ends, the log files are available at:

Oracle Linux:

```
/var/log/utsetup.year_month_date_hour:minute:second.log
```

Oracle Solaris:

```
/var/adm/log/utsetup.year_month_date_hour:minute:second.log
```

The values in the file names reflect a time stamp of when the command was started. Check these files for notices of problems.

See [Section 3.2.12, “Installation \(utinstall\) Error Messages”](#) for a listing of `utinstall` error messages.

6. Repeat steps 3 through 5 for each secondary server if in a failover group.

If you choose the HA group (failover group) configuration during the `utsetup` installation of the primary server, you can copy the generated response files to each secondary server and use the `utsetup` command to replicate the failover group configuration. For example, you can save the response files to the secondary's server's `/tmp` directory and then use the `utsetup -a -D /tmp` command.

This strategy reduces the time needed to re-enter the same configuration information and can minimize configuration errors. See [Section 3.2.8, “How to Clone a Sun Ray Server”](#) for details.

7. Add the Sun Ray server's host name or IP address to the `sunray-config-servers` and `sunray-servers` DNS entries, which will make the server available to clients for firmware updates and Sun Ray sessions, respectively.

See [Chapter 2, *Planning a Sun Ray Network Environment*](#) for details.

8. To configure any site-wide settings for the Sun Ray Clients, such as the automatic poweroff feature, you need to update the `.parms` file for each client. You need to do this on each designated firmware server.

See [Section 13.2.1, “How to Centralize Sun Ray Client Configurations \(.parms\)”](#) for details.

9. Reboot the Sun Ray Clients to download and update to the new Sun Ray Operating Software (firmware) provided by the Sun Ray server.



Note

When installing Sun Ray Operating Software on Sun Ray 3 Series Clients, the smart card LED will blink for approximately 40 seconds as the smart card controller firmware is being updated. This is normal.

10. If you plan to use the Windows connector, install the Windows Connector Components on your designated Windows Server. See [Section 3.2.7, “How to Install the Windows Connector Components on a Windows System”](#) for details.

3.2.6. Post-Installation Configuration

This section includes various configuration procedures that you can perform on the Sun Ray server after installing the Sun Ray Software.

3.2.6.1. How to Install the JDS Integration Package (Oracle Solaris 10)



Note

The Java Desktop System (JDS) integration package is a deprecated feature and will be removed in a later release.

The Java Desktop System (JDS) integration package, which delivers a CLI called `uttscwrap` that improves integration of the Windows connector with the JDS desktop on Oracle Solaris 10. For more information about the `uttscwrap` command, see [Section 17.3.2, “How to Start a Windows Session Within Java Desktop System \(Oracle Solaris 10\)”](#).

1. Change to the media pack location where the JDS integration package is located.

```
# cd media_pack_directory/Supplemental/JDS_Integrator/Solaris*/arch/Packages
```

2. Install the JDS integration package (`SUNWuttscwrap`).

```
# pkgadd -d .
```

The `uttscwrap` command is installed in the `/opt/SUNWuttscwrap/bin` directory.

3.2.6.2. How to Configure a Headless Sun Ray Server (Oracle Linux)

If you plan to use a headless Sun Ray server running Oracle Linux and Sun Ray Clients using GNOME Display Manager (GDM), then this configuration will generate errors on the Sun Ray Client and consume CPU processes. The errors occur because GDM assumes that the console display is present and GDM will continually attempt to (and fail to) service a non-existent console device.

The workaround is to add the `-no-console` option to the `preadm` command in the Sun Ray server's `/etc/inittab` file:

```
x:5:respawn:/etc/X11/prefdm --nodaemon --no-console
```

This workaround is not required for Oracle servers, since they have the Integrated Lights Out Manager (ILOM) Service Processor that provides a virtual console.

3.2.6.3. How to Limit Administrative Privileges for Non-root Users (Oracle Linux)

Many Oracle Linux systems come configured with liberal administrative privileges for non-root users. These privileges should not be made available to users who log in using a Sun Ray Client.

To limit administrative access, do the following:

- Review the man pages for `pam_console`, `console.perms`, and `console.apps`.
- Edit the `/etc/security/console.perms` file to remove display numbers from the definition of console. If a definition exists for `xconsole`, it should be removed.

For example, a line that reads:

```
<console>=tty[0-9][0-9]* vc/[0-9][0-9]* :[0-9]?[0-9] :[0-9]
```

should instead read:

```
<console>=tty[0-9][0-9]* vc/[0-9][0-9]*
```

And a line such as the following example should be removed:

```
<xconsole>=: [0-9]?[0-9] :[0-9]
```

3.2.7. How to Install the Windows Connector Components on a Windows System

The Windows connector feature enables you to provide Windows remote desktop services on Sun Ray Clients. Sun Ray Software provides a number of Windows connector components that you should install on a Windows system to improve the performance and functionality of the remote desktop services.



Note

By default, remote desktop services is not enabled on a Windows system, so you must specifically enable it. See the Windows documentation for details.

This procedure provides the steps to install the Windows connector components on a Windows system:

- USB redirection - Enables access to USB devices connected to a Sun Ray Client from a Windows session.
- Multimedia redirection - Enhanced performance for Windows Media Player (Windows XP and Windows Server 2003 R2).
- Adobe Flash acceleration - Enhanced playback capabilities for Adobe Flash content.
- Audio/video synchronization - Enhanced audio and video synchronization for multimedia content (Windows XP and Windows Server 2003 R2).
- Audio input - Enables audio recording on a Sun Ray Client from a Windows session (Windows XP and Windows Server 2003 R2).
- Client Information Agent - Enables the client name to be updated across hotdesking and to provide the ability to execute actions on disconnects and reconnects of a Windows session.

If you want to install the Sun Ray Connector Windows components by using the *.msi files, you can use a `srs-winstaller.exe /S /D=c:path` to extract the *.msi files from the `srs-wininstaller.exe` executable.



Note

To bypass the installation UI, you can run `srs-winstaller /S` from the command line.



Note

The Adobe Flash acceleration and Audio/video synchronization components require hardware that supports the Windows Performance Counter API. If the Windows Performance Counter API is not working properly, the components might fail to load or behave unexpectedly. In one known example, this problem occurs when a computer has the AMD Cool'n'Quiet technology enabled in the BIOS, which is documented in <http://support.microsoft.com/kb/895980>.

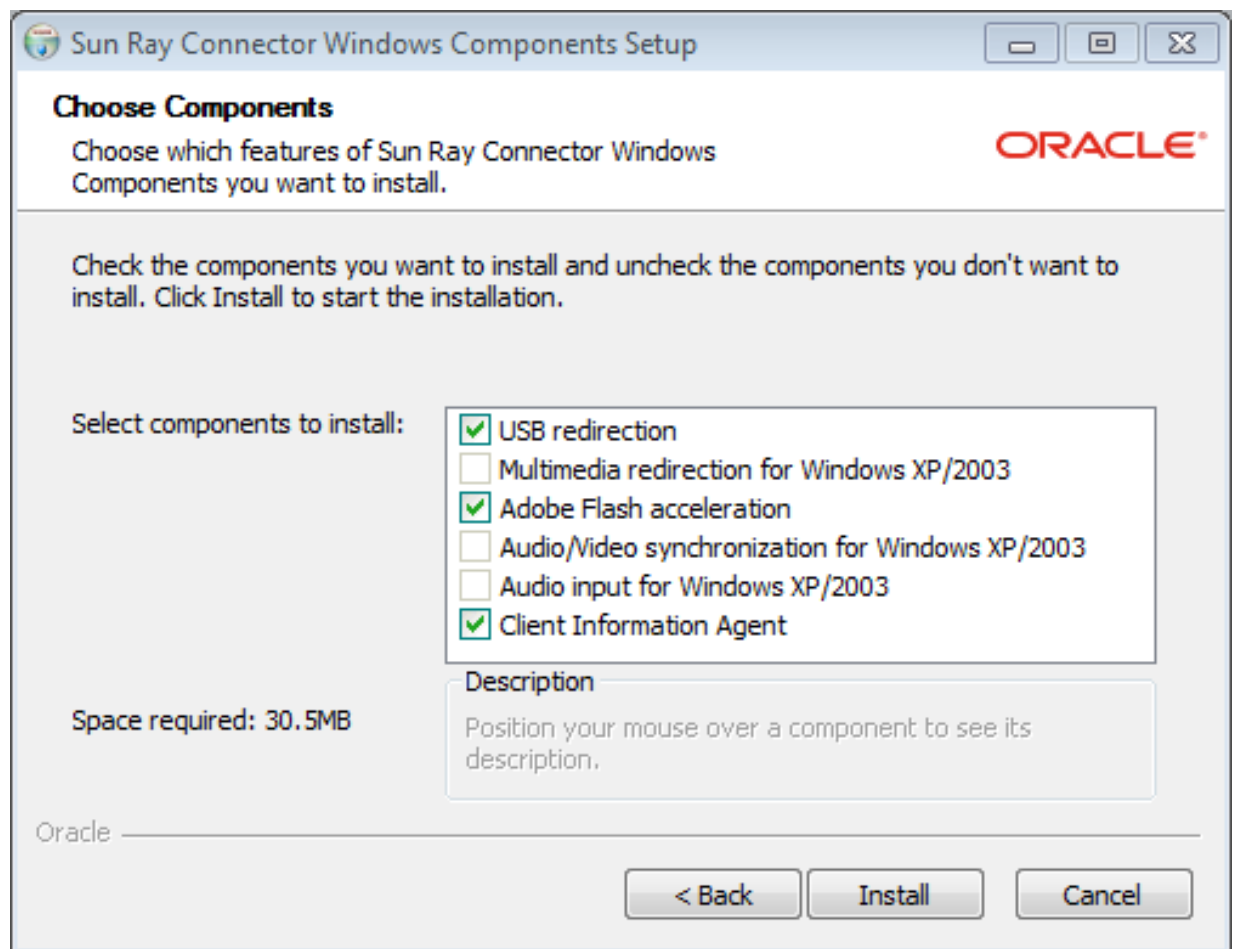
Steps

1. Log in to the Windows system as Administrator.
2. If you plan to install the USB redirection component on a Virtual Machine (VM), you must add USB drivers on some VMs if they do not provide drivers by default. See [Section 17.6.5, “How to Add USB Drivers to a Virtual Machine”](#) for details.
3. Make sure the Windows system has access to the Sun Ray Connector Windows Components installer in the unzipped Sun Ray Software media pack.

`media_pack_image/Components/20-SRWC/Content/Sun_Ray_Connector_Windows_Components_2.6`

4. Copy the `srs-winstaller.exe` file to the Windows system.
5. Double-click the **srs-winstaller** icon to start the Sun Ray Connector Windows Components Setup Wizard.
6. Review the License Agreement and click **I Agree**.
7. Choose which components you want to install and click **Install**.

Figure 3.1. Windows Components Setup Window



8. Click **Finish** once the installation has finished.

Restart the Windows system if instructed.

9. Go to the following sections (next steps) based on the features you installed.

- [Multimedia Redirection - Next Steps](#)
- [Adobe Flash Acceleration - Next Steps](#)
- [Audio/Video Synchronization - Next Steps](#)
- [USB Redirection - Next Steps](#)

3.2.7.1. Multimedia Redirection - Next Steps

The multimedia redirection component does not include an audio/video demuxer for MPEG-2 and H.264 video streams. To ensure that video is accelerated, download and install a third-party or freeware solution, such as the MatroskaSplitter freeware.

3.2.7.2. Adobe Flash Acceleration - Next Steps

For Adobe Flash videos, users must enable the **Third party browser extensions** option in Internet Explorer, which is located in the Advanced tab of **Tools > Internet Options**.

3.2.7.3. Audio/Video Synchronization - Next Steps



Note

For audio to work properly, the Sun Ray audio driver must be set as the default. If users have changed their default audio driver, they must perform the following procedure to make the Sun Ray audio driver the default.

1. From the Windows Desktop, choose **Settings > Control Panel**.
2. Click **Sounds & Audio Devices**.
3. Click the **Audio** tab.
4. If the Sun Ray RDP Audio Driver is not the default, select it and click **Apply**.
5. Close your browser and reopen it.

3.2.7.4. USB Redirection - Next Steps

To verify that USB redirection is installed properly, see [Figure 17.8, “Verifying USB Redirection in Windows Device Manager”](#).

To verify that USB redirection is working from a Windows connector session, see [Section 17.6.6.1, “How to Verify that USB Redirection is Active”](#).

3.2.7.5. How to Repair the Windows Connector Components

The `srs-winstaller.exe` executable is an archive, which extracts two `.msi` files into a temporary directory and then transfers control to the Microsoft Windows installer. If the installed Windows Connector Components software needs to be repaired, the `.msi` files may no longer be available.

There are two ways to repair the Windows Connector Components software. The easiest method is to uninstall the software using the add/remove interface and then reinstall using the procedure above.

The other repair method is to select the Windows Connector Components in the add/remove interface and click Repair. If the repair process requests the location of the `.msi` files, use the following procedure to extract the files from the `srs-winstaller.exe` executable:

1. Create a new directory with no blanks in the path or name. The example in this procedure uses `c:\srwc`.
2. In a command shell window, run the following command:

```
srs-winstaller.exe /S /D=c:\srwc
```

The `.msi` files are now available in the `c:\srwc` directory.

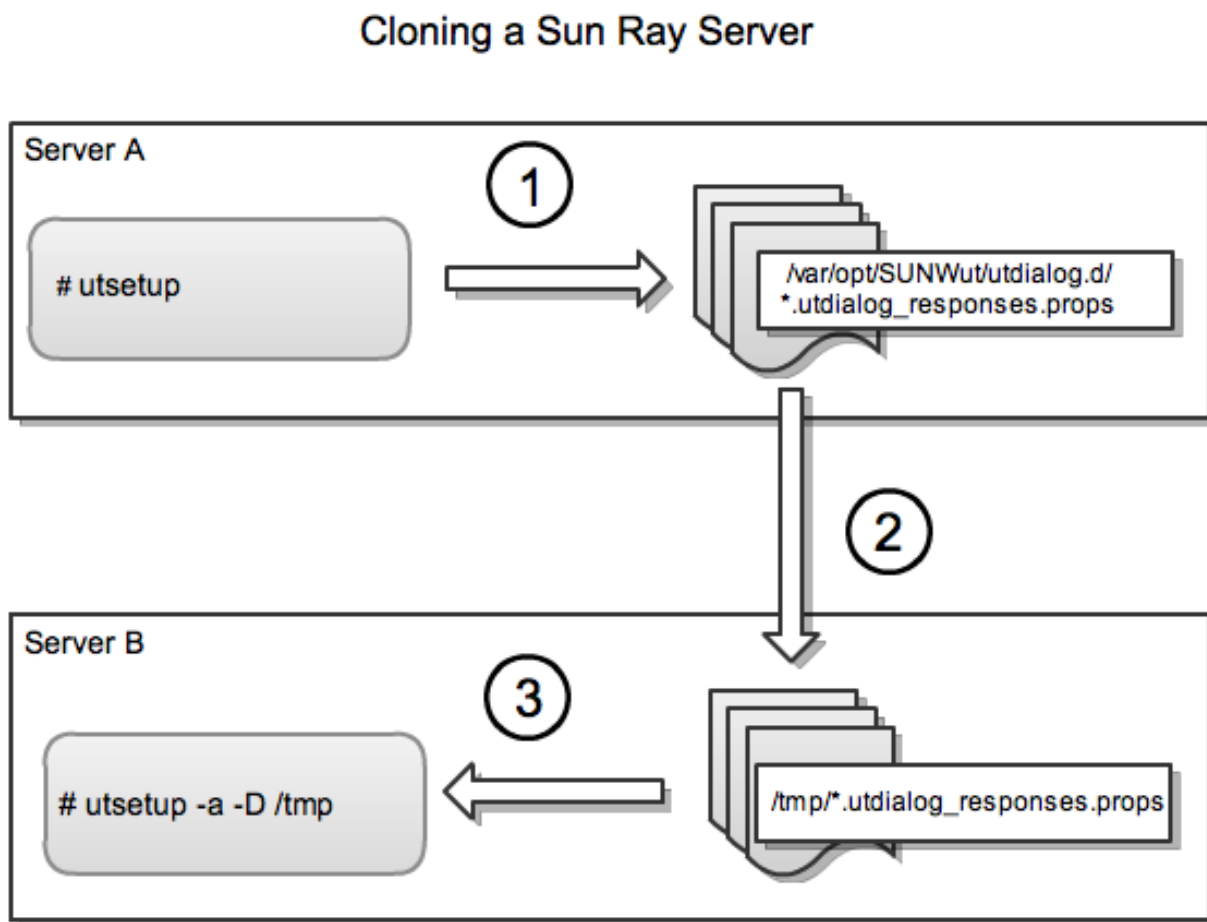
3.2.8. How to Clone a Sun Ray Server

The `utsetup` command enables you to install and configure a Sun Ray server with the Sun Ray Software. You can then use the created response files to install and configure other servers. See [Section 3.2.3](#),

[“Automating Sun Ray Software Installations”](#) for more information about automating Sun Ray Software installations.

Figure 3.2, “Cloning a Sun Ray Server” shows how you can use the `utsetup` command to clone Sun Ray servers.

Figure 3.2. Cloning a Sun Ray Server



Note

The `*.utdialog_responses.props` files may contain passwords that a malicious user could potentially decode if read, so make sure to use sufficient security precautions when copying them. For example, the files should have secure permissions (not readable by group or "other"), and you should remove the files from their temporary locations after you complete the configuration.

1. From the unzipped Sun Ray Software media pack, run the `utsetup` command to install and configure a Sun Ray server.

The responses you provide are saved in the `/var/opt/SUNWut/utdialog.d/*.utdialog_responses.props` files.

2. Copy the response files to another server. In this example, the files are copied to Server B's `/tmp` directory.
3. From the unzipped Sun Ray Software media pack, run the `utsetup -a -D /tmp` command to clone the installation and configuration on the server that you created in Step 1.

For members of the same failover group, you can typically apply the configuration without changes. For a new failover group, you may need to edit the response files to create a new configuration (for example to specify a new primary and secondaries for replication). The `utdialog_responses.props(5)` man page gives detail about the format of the files.

3.2.9. How to Install and Configure a Sun Ray Server With Default Settings

Another way to use the `utsetup` command is to quickly install and configure a server with the default settings.

1. Install, configure, and activate the basic Sun Ray Software product features for standalone use (no failover group configuration) with a minimum of user interaction required (JRE 1.6 must be installed at `/usr/java` before proceeding).

```
utsetup -d
```

2. Complete the configuration:

- a. Optionally configure the system in a failover group (if you did not specify identical administration passwords with the `utsetup` command for each system, then you must run the `utpw` command at this time).

```
# utgroupsig
```

```
# utreplica -p secondary-server1 [secondary-server2...]
```

or

```
# utreplica -s primary-server
```

- b. Optionally configure the kiosk functionality.

```
# utconfig -k -d
```

- c. Optionally configure the browser web administration interface (Apache Tomcat must be installed at `/opt/apache-tomcat` before proceeding).

```
utconfig -w -d
```

- d. Configure the Sun Ray network services.

```
# utadm -L on
```

or

```
# utadm -A subnet
```

or

```
# utadm -a interface
```

- e. Start the Sun Ray services

```
# utstart
```

3.2.10. How to List the Current Sun Ray Software Version

This procedure lists the current version of Sun Ray Software running on the Sun Ray server.

1. Log in as the superuser of the Sun Ray server.
2. List the current version.

```
# /opt/SUNWut/sbin/utrelease
```

3.2.11. How to Remove Sun Ray Software

To remove Sun Ray Software in its entirety, follow this procedure.

1. Log in as the superuser of the Sun Ray server.
2. Open a shell window and change to the `/opt/SUNWut/sbin` directory.

```
# cd /opt/SUNWut/sbin
```

3. If you are removing Sun Ray Software from a server in a failover group, disable Sun Ray Client firmware updates.

See [Section 14.13, “How to Disable All Sun Ray Client Firmware Updates”](#) for details.

4. Remove the replication configuration.

```
# ./utreplica -u
```

5. Remove the Sun Ray network interfaces.

```
# ./utadm -r
```

6. Unconfigure Sun Ray Software.

```
# ./utconfig -u
```

Answer **y** to all of the prompts.

7. Uninstall Sun Ray Software.

```
# cd /  
# /opt/SUNWut/sbin/utinstall -u
```

Answer **y** to all of the prompts.

8. Reboot the system.
9. Repeat the steps in this procedure for all remaining Sun Ray servers.

3.2.12. Installation (utinstall) Error Messages

If during an installation, upgrade, or uninstall the `utinstall` script returns an error, refer to the following table for assistance.

3.2.12.1. All Installations

Message	Meaning	Resolution
<code>utinstall: fatal, media-dir is not a valid directory.</code>	You called the <code>-d</code> option, but <code>media-dir</code> is incomplete.	The <code>media-dir</code> directory requires relevant patches and packages for

Message	Meaning	Resolution
xxxxxx not successfully installed	Might occur for the installation of any application or patch if relevant packages have not been properly installed.	installation. The <code>media-dir</code> directory includes the Sun Ray directory.
A different version x.x of product has been detected. The other-product Software is only compatible with product y.y. You must either upgrade or remove the current product installation before proceeding. Exiting ...	Some of the applications provided with Sun Ray Software are compatible only with certain versions of other applications.	Compatible and necessary applications are included with Sun Ray Software. Remove older versions, then run the <code>utinstall</code> script again.
error, no Sun Ray software packages installed.	None of the Sun Ray components are installed on this system.	No action is required as the product is not installed.
The following files were not successfully replaced during this upgrade. The saved copies can be found in <code>directory</code>	Some files were not properly replaced as part of the upgrade.	Manually copy the listed files from the directory, overwriting the newer files if applicable.
Space Required and Space Available table	Not enough disk space was allocated for partition. Repartition the disk and run <code>utinstall</code> again.	

3.2.12.2. Oracle Linux Installations

Message	Meaning	Resolution
The following packages were not successfully removed xxxxxx ...	The packages listed have not been properly removed.	Use the <code>rpm -e</code> command to remove each listed rpm manually, then run <code>utinstall -u</code> again.
Removal of product was not successfully completed. See log file for more details.	Removal of Sun Ray Software was incomplete.	Check the log file for the package that started the problem and manually remove it with the <code>rpm -e</code> command, then run <code>utinstall -u</code> again.

3.2.12.3. Oracle Solaris Installations

Message	Meaning	Resolution
Cannot open for read <code>admin-file</code>	The <code>admin_default</code> file is unreadable, or you called the <code>-a</code> option and the <code>admin-file</code> is unreadable.	Verify that the installation administration file exists (<code>admin_default</code> or other) and the permissions are correct.
For SPARC platforms: SunOS release is x.x, valid releases are: 10	You are attempting to install Sun Ray Software onto an Oracle Solaris	Upgrade to the supported version 10 of the Oracle Solaris OS before installing Sun Ray Software.

Message	Meaning	Resolution
	version that does not support the Sun Ray Software release.	
For x86 platforms: SunOS release is x.x, valid releases are: 10	You are not running a valid OS release for this platform.	Upgrade to the supported version 10 of the Oracle Solaris OS before installing Sun Ray Software.
Please clean up the directory /var/tmp/SUNWut.upgrade before rerunning utinstall.	Other unrelated files were found in the <code>preserve</code> directory.	Remove unrelated files from the directory.
Please remove the existing preserved file, <code>preserved_tarfilename</code> , before rerunning utinstall.	You decided not to restore from the indicated tar file.	Remove the tar file before running <code>utinstall</code> again.
utpreserve: unable to preserve data. Error while creating archive file	The <code>utinstall</code> script failed to preserve existing configuration files.	Either exit and manually preserve these files or just continue.
The following packages were not successfully removed xxxxxxx ...	The packages listed have not been properly removed.	Use the <code>pkgrm</code> command to remove each listed package manually, then run <code>utinstall -u</code> again.
Removal of product was not successfully completed. See log file for more details.	Removal of Sun Ray Software was incomplete.	Check the log file for the package that started the problem and manually remove it with the <code>pkgrm</code> command, then run <code>utinstall -u</code> again.

3.3. Configuring Oracle Solaris 11 Trusted Extensions

This section provides the procedure that needs to be done when using Sun Ray Software on Oracle Solaris 11 Trusted Extensions. For more information, refer to the [Oracle Solaris 11 Trusted Extensions Configuration and Administration Guide](#).

Oracle Solaris 11 uses zones to permit multiple virtualized operating system environments to coexist in a single instance of Oracle Solaris 11, allowing processes to run in isolation from other activity on the system for added security and control. Sun Ray Software is supported only in the global zone.

Based on your Sun Ray environment, perform the following procedure as root from ADMIN_LOW (global zone).

3.3.1. How to Configure Sun Ray Software on Oracle Solaris 11 Trusted Extensions

This procedure is required to configure Sun Ray Software on Oracle Solaris 11 Trusted Extensions. The labeled zone named `public` is used in examples throughout this procedure.

1. Become root from ADMIN_LOW (global zone).
2. Configure the following Multilevel ports for the global zone.
 - a. Run the `txzonemgr` script:

```
# txzonemgr
```

- b. Choose **Global Zone > Configure Multilevel Ports > Add MLP-shared-tcp**
- c. Add the following Multilevel ports:
 - 4120 - Smart card service daemon (pcscd)
 - 6000-6050 - Xserver ports (if more than 50 sessions are needed, increase this port range accordingly.)
 - 7007 - Session manager daemon (utsessiond)
 - 7010 - Authentication manager daemon (utauth-cb)
 - 7012 - Data store daemon (utds)
 - 7014 - Windows connector daemon (uttscpd)
 - 7015 - Audio daemon
3. If you are providing Windows remote desktops through the Windows connector, enable access to each system through the labeled zone:
 - a. Add an entry for each Windows system to the `/etc/security/tsol/tnrhdb` file:


```
windows-IP:labeled-zone
```

The following example enables access to a Windows system with an IP address of 10.178.231.24 from the `public` zone:

```
10.178.231.24:public
```
 - b. Restart network services:


```
# svcadm restart tnctl
```
4. (Optional) For TLS peer verification to work, make sure the CA certificates to be trusted are available under the `/etc/sfw/openssl/certs` folder in each labeled zone.
5. Loopback mount the following directories and applications for each labeled zone. The following example shows how to do this for the `public` zone.



Note

Setting up a loopback mount for `libmllib.so` and `libmllib.so.2` is required only for SPARC-based Sun Ray servers

```
# zoneadm -z public halt
# zonecfg -z public

zonecfg:public> add fs
zonecfg:public:fs> set dir=/opt/SUNWut
zonecfg:public:fs> set special=/opt/SUNWut
zonecfg:public:fs> set type=lofs
zonecfg:public:fs> end
zonecfg:public> add fs
zonecfg:public:fs> set dir=/etc/opt/SUNWut
zonecfg:public:fs> set special=/etc/opt/SUNWut
zonecfg:public:fs> set type=lofs
zonecfg:public:fs> end
zonecfg:public> add fs
```

```
zonecfg:public:fs> set dir=/usr/lib/libpcsc-lite.so
zonecfg:public:fs> set special=/usr/lib/libpcsc-lite.so
zonecfg:public:fs> set type=lofs
zonecfg:public:fs> end
zonecfg:public> add fs
zonecfg:public:fs> set dir=/usr/lib/libpcsc-lite.so.1
zonecfg:public:fs> set special=/usr/lib/libpcsc-lite.so.1
zonecfg:public:fs> set type=lofs
zonecfg:public:fs> end
zonecfg:public> add fs
zonecfg:public:fs> set dir=/etc/opt/SUNWuttsc
zonecfg:public:fs> set special=/etc/opt/SUNWuttsc
zonecfg:public:fs> set type=lofs
zonecfg:public:fs> end
zonecfg:public> add fs
zonecfg:public:fs> set dir=/opt/SUNWuttsc
zonecfg:public:fs> set special=/opt/SUNWuttsc
zonecfg:public:fs> set type=lofs
zonecfg:public:fs> end
zonecfg:public> add fs
zonecfg:public:fs> set dir=/usr/lib/libmllib.so
zonecfg:public:fs> set special=/usr/lib/libmllib.so
zonecfg:public:fs> set type=lofs
zonecfg:public:fs> end
zonecfg:public> add fs
zonecfg:public:fs> set dir=/usr/lib/libmllib.so.2
zonecfg:public:fs> set special=/usr/lib/libmllib.so.2
zonecfg:public:fs> set type=lofs
zonecfg:public:fs> end
zonecfg:public> exit

# zoneadm -z public boot
```

6. Reboot the Sun Ray server.

```
# reboot
```

3.4. Configuring Oracle Solaris 10 Trusted Extensions

This section provides all the procedures that may need to be done when using Sun Ray Software on Oracle Solaris 10 Trusted Extensions. For more information, refer to the [Oracle Solaris 10 8/11 Trusted Extensions Administrator's Procedures](#).

Oracle Solaris 10 uses zones to permit multiple virtualized operating system environments to coexist in a single instance of Oracle Solaris, allowing processes to run in isolation from other activity on the system for added security and control. Sun Ray Software is supported only in the global zone.

Based on your Sun Ray environment, perform the following procedures as root from ADMIN_LOW (global zone).

3.4.1. How to Configure a Private Network on Oracle Solaris 10 Trusted Extensions

This procedure is required if your Sun Ray server is configured on a private network. See [Chapter 19, Alternate Network Configurations](#) for more information.

Use the Solaris Management Console (SMC) Security Templates to assign the `cipso` template to the Sun Ray server. Assign all other Sun Ray devices on the network an `admin_low` label. The `admin_low` template is assigned to the range of IP addresses you are planning to use in the `utadm` command.

The `/etc/security/tsol/tnrhdb` file should contain the following entries when you finish:

```
192.168.128.1:cipso
192.168.128.0:admin_low
```

1. Become root from ADMIN_LOW (global zone).
2. Start the Solaris Management Console (SMC).

```
# smc &
```

3. Make the following selections:
 - a. In the SMC, select **Management Tools > Select hostname:Scope=Files, Policy=TSOL**.
 - b. Select **System Configuration > Computers and Networks > Security Templates > cipso**.
 - c. From the menu bar, choose **Action > Properties > Hosts Assigned to Template**.
 - d. Select **Host** and type the IP Address of the Sun Ray interconnect (for example, 192.168.128.1).
 - e. Click **Add** and then **OK**.
 - f. Select **System Configuration > Computers and Networks > Security Families > admin_low**.
 - g. From the menu bar, choose **Action > Properties > Hosts Assigned to Template**.
 - h. Select **Wildcard**.
 - i. Type the IP Address of the Sun Ray Interconnect Network (192.168.128.0).
 - j. Click **Add** and then **OK**.
4. Assign all Sun Ray servers in the failover group a **cipso** label.
 - a. Select **System Configuration > Computers and Networks > Security Families > cipso**.
 - b. From the menu bar, choose **Action > Properties > Hosts Assigned to Template**.
 - c. Select **Host** and type the IP Address of the other Sun Ray server.
 - d. Click **Add** and then **OK**.
5. Reboot the Sun Ray server.

```
# /usr/sbin/reboot
```

3.4.2. How to Configure Shared Multilevel Ports (MLP) for Sun Ray Services

A shared multilevel port has to be added to the global zone for Sun Ray services in order to have access from a labeled zone.

1. Become root from ADMIN_LOW (global zone).
2. Start the Solaris Management Console (SMC).

```
# smc &
```

3. Go to Management Tools.
4. Select **hostname:Scope=Files, Policy=TSOL**.
5. Select **System Configuration > Computers and Networks > Trusted Network Zones > global**.

6. From the menu bar, choose **Action > Properties**.
7. Click **Add** under **Multilevel Ports for Shared IP Addresses**.
8. Add 7007 as **Port Number**, select **TCP** as **Protocol**, and click **OK**.
9. Repeat the previous step for ports 4120, 7010, and 7015.
10. Restart network services by running the following command:

```
# svcadm restart svc:/network/tnctl
```

11. Verify that these ports are listed as shared ports by running the following command:

```
# /usr/sbin/tninfo -m global
```

12. Reboot the Sun Ray server.

```
# /usr/sbin/reboot
```

3.4.3. How to Increase the Number of X Server Ports

The default entry in `/etc/security/tsol/tnzonecfg` makes three displays available (6001-6003). Increase the number of available X server ports per requirements.

1. Become root from ADMIN_LOW (global zone).
2. Start the Solaris Management Console (SMC).

```
# smc &
```

3. Go to Management Tools.
4. Select **hostname:Scope=Files, Policy=TSOL** option.
5. Select **System Configuration > Computers and Networks > Trusted Network Zones > global**.
6. From the menu bar, choose **Action > Properties**.
7. Under **Multilevel Ports for Zone's IP Addresses**, select **6000-6003/tcp**.
8. Click **Remove**.
9. Choose **Add > Enable Specify A Port Range**.
10. Type **6000** in **Begin Port Range Number** and **6050** (for 50 displays) in **End Port Range Number**.
11. Select **TCP** as the **Protocol**.
12. Click **OK**.
13. Reboot the Sun Ray server.

```
# /usr/sbin/reboot
```

3.4.4. How to Configure the Windows Connector on Oracle Solaris Trusted Extensions

This procedure describes how to configure the Windows connector on Oracle Solaris Trusted Extensions.

For the Windows connector to function properly on a Oracle Solaris Trusted Extensions server, the Windows terminal server must be made available at the desired level.

1. As superuser, open a shell window on the Sun Ray server.

To avoid errors that can occur if user environment settings are carried forward, use the following command:

```
% su - root
```

2. Make a Windows system available to the `public` template.

- a. Start the Solaris Management Console.

```
# smc &
```

- b. Make the following selections under Management Tools:

- i. Select **hostname:Scope=Files, Policy=TSOL**.
- ii. Select **System Configuration > Computers and Networks > Security Templates > public**.

- c. Choose **Action > Properties > Hosts Assigned to Template**.

- d. Select **Host**.

- e. Type the IP Address of the Windows system, for example, 10.6.100.100.

- f. Click **Add**.

- g. Click **OK**.

3. Configure port 7014 as a shared multilevel port for the `uttscpd` daemon.

- a. If the Solaris Management Console is not already running, start it:

```
# smc &
```

- b. Select **hostname:Scope=Files, Policy=TSOL**.

- c. Select **System Configuration > Computers and Networks > Trusted Network Zones > global**.

- d. Choose **Action > Properties**.

- e. Enable ports by clicking **Add** under **Multilevel Ports for Shared IP Addresses**.

- f. Add 7014 as **Port Number**, select **TCP** as the **Protocol**, and click **OK**.

- g. Restart network services.

```
# svcadm restart svc:/network/tnctl
```

- h. Verify that this port is listed as a shared port.

```
# /usr/sbin/tninfo -m global
```

4. Create entries for the `uttscpd` daemon in each local zone.

The `/etc/services` file entry for the SRWC proxy daemon is created automatically in the global zone at configuration time. Corresponding entries need to be created in the local zones.

These entries can be created manually or by loopback-mounting the global zone `/etc/services` file into the local zones for read access.

To create this entry manually, insert the following entry in the local zone file.

```
uttscpd 7014/tcp # SRWC proxy daemon
```

5. Loopback mount the `/etc/opt/SUNWutts` directory in each local zone. The following example shows how to do this for the local zone named `public`.

```
# zoneadm -z public halt
# zonecfg -z public

zonecfg:public> add fs
zonecfg:public:fs> set dir=/etc/opt/SUNWutts
zonecfg:public:fs> set special=/etc/opt/SUNWutts
zonecfg:public:fs> set type=lofs
zonecfg:public:fs> end

# zoneadm -z public boot
```

6. (Optional) For TLS peer verification to work, make sure the CA certificates to be trusted are available under the `/etc/sfw/openssl/certs` folder in each local zone.
7. Reboot the Sun Ray server.

```
# /usr/sbin/reboot
```

3.5. Upgrading

This section provides instructions on how to upgrade a previously installed Sun Ray server.

3.5.1. Installing Firmware Before Sun Ray Software Upgrade

As described in [Section 3.2.4, “Installing Firmware Before Sun Ray Software Installation”](#), it is recommended that you install the latest firmware on the existing Sun Ray server before performing an upgrade. The firmware is not provided with the Sun Ray Software media pack.

The current upgrade procedure provides steps on how to download and install the firmware. To update client firmware outside of the Sun Ray Software upgrade process, refer to [Section 14.3, “How to Update Firmware on Sun Ray Clients”](#).

3.5.2. How to Upgrade Sun Ray Software

This procedure describes how to upgrade Sun Ray Software on an existing Sun Ray server. This procedure relies on a specific Sun Ray configuration. See [Chapter 2, *Planning a Sun Ray Network Environment*](#) for details.

Note the following information before performing the upgrade.

- Make sure the operating system on the Sun Ray server complies with the Sun Ray Server 5.4 requirements listed in [Section 3.1.1, “Operating System Requirements”](#). If not, you'll have to upgrade the operating system on the Sun Ray server as part of the Sun Ray Software upgrade.
- Upgrades from Sun Ray Software 5.2 or later are supported with Sun Ray Software 5.4. You can upgrade directly to the latest Sun Ray Software 5.4.x release update (if available), which means you do not have to upgrade to Sun Ray Software 5.4 first before upgrading to Sun Ray Software 5.4.x.
- You cannot migrate a Sun Ray server configuration to a hardware platform of a different Instruction Set Architecture. For example, you cannot migrate an existing SPARC-based Sun Ray server configuration to a new x86-based Sun Ray server.

- You cannot migrate a Sun Ray server configuration to a different operating system, for example, from Oracle Linux to Oracle Solaris. Although, you can upgrade between major releases of the same operating system when specified, such as Oracle Solaris 10 to Oracle Solaris 11.
- Although not necessary in most circumstances, you should preserve the configuration data on the Sun Ray server and copy the backup file to another location. See [Section 3.5.4, “How to Preserve Sun Ray Software Configuration Data”](#) for details.

Steps

1. Inform users about the upgrade.

Before you upgrade Sun Ray Software, inform your users of your plans, and have them terminate their sessions. An effect of the upgrade procedure is that all active and suspended sessions are lost.

2. If you are upgrading Sun Ray servers in a failover group, consider ways to reduce downtime.

See [Section 3.5.3, “Planning Upgrades Using Failover Groups”](#) for details.

3. Become superuser on the Sun Ray server.

To avoid installation script errors that can occur if user environment settings are carried forward, use the following command:

```
% su - root
```

4. List the current Sun Ray network configuration and retain the information. You need to reconfigure the Sun Ray network after the upgrade.

```
# /opt/SUNWut/sbin/utadm -l
```

5. If necessary, upgrade the operating system on the Sun Ray server to meet the requirements listed in [Section 3.1.1, “Operating System Requirements”](#).

Upgrading Oracle Linux 5.x to 5.8

- a. Preserve the configuration data on the Sun Ray server and copy the backup file to a safe location. See [Section 3.5.4, “How to Preserve Sun Ray Software Configuration Data”](#) for details.
- b. Uninstall the Sun Ray Software on the Sun Ray server. See [Section 3.2.11, “How to Remove Sun Ray Software”](#) for details.



Note

If the `utkiosk` group is still configured on the system, remove it. Otherwise, the kiosk mode user account configuration will fail during the Sun Ray Software upgrade.

- c. Upgrade Oracle Linux on the Sun Ray server.
- d. If necessary, copy the Sun Ray server configuration data backup file created earlier, `/var/tmp/SUNWut.upgrade/preserve_version.tar.gz`, to the same location on the upgraded Sun Ray server. The Oracle Linux upgrade should have retained this file.
- e. Go to Step 6.

Upgrading Oracle Linux 5.x to 6.3

- a. Preserve the configuration data on the Sun Ray server and copy the backup file to a safe location. See [Section 3.5.4, “How to Preserve Sun Ray Software Configuration Data”](#) for details.
- b. Upgrade Oracle Linux on the Sun Ray server.

In-place upgrades are not supported between major Oracle Linux releases, so it is recommended that you perform a fresh install of Oracle Linux 6.3 on the Sun Ray server after backing up the existing system. Refer to the Oracle Linux documentation for details on upgrading between major Oracle Linux releases.

- c. Copy the Sun Ray server configuration data backup file created earlier, `/var/tmp/SUNWut.upgrade/preserve_version.tar.gz`, to the same location on the upgraded Sun Ray server.
- d. Go to Step 6.

Upgrading Oracle Solaris 10

- a. (Optional) Preserve the configuration data on the Sun Ray server and move the backup file to a safe location. See [Section 3.5.4, “How to Preserve Sun Ray Software Configuration Data”](#) for details.

This step is not necessary because the Oracle Solaris 10 upgrade will not affect the Sun Ray Software configuration data. However, backing up data before performing an operating system upgrade is always good practice.

- b. Upgrade Oracle Solaris on the Sun Ray server.
- c. Go to Step 6.

Upgrading Oracle Solaris 10 to Oracle Solaris 11

- a. Preserve the configuration data on the Sun Ray server and copy the backup file to a safe location. See [Section 3.5.4, “How to Preserve Sun Ray Software Configuration Data”](#) for details.
- b. Upgrade Oracle Solaris on the Sun Ray server.

There is no upgrade program to upgrade Oracle Solaris 10 to Oracle Solaris 11. You must perform a fresh install of Oracle Solaris 11 on the Sun Ray server after backing up the existing system. Refer to the Oracle Solaris 11 documentation for details on upgrading to Oracle Solaris 11.

- c. Copy the Sun Ray server configuration data backup file created earlier, `/var/tmp/SUNWut.upgrade/preserve_version.tar.gz`, to the same location on the upgraded Sun Ray server.
- d. Go to Step 6.

6. Download and unzip the Sun Ray Software 5.4 media pack and make it accessible to the Sun Ray server.

See <http://www.oracle.com/technetwork/server-storage/sunrayproducts/downloads/index.html>

7. (Optional) Download and unzip the latest Sun Ray Operating Software (firmware) and make it accessible to the Sun Ray server.

See <http://www.oracle.com/technetwork/server-storage/sunrayproducts/downloads/index.html>

If you decide to install and configure client firmware after the upgrade, refer to [Section 14.3, “How to Update Firmware on Sun Ray Clients”](#).

8. Disable all Sun Ray Client firmware updates until all the servers in a failover group are upgraded.

See [Section 14.13, “How to Disable All Sun Ray Client Firmware Updates”](#) for details.

9. If you downloaded the latest Sun Ray Operating Software (firmware), change directory to the unzipped firmware directory and update the current firmware to make it available to the Sun Ray Software installation upgrade.

```
# ./utfwinstall
```

The `utfwinstall` script overwrites the existing firmware installed on the Sun Ray server.

10. Change directory to the unzipped Sun Ray Software media pack and upgrade Sun Ray Software on the Sun Ray server.

```
# ./utsetup
```

The `utsetup` script preserves the current Sun Ray Software configuration data on the Sun Ray server, upgrades Sun Ray Software to the new version, and then restores the Sun Ray Software configuration data after the upgrade. If you had to create a `/var/tmp/SUNWut.upgrade/preserve_version.tar.gz` backup file and copy it to the newly installed OS as described in Step 5, the `utsetup` script will prompt you to use the backup file to restore the Sun Ray Software configuration data.

When the script ends, a log file is available at:

Oracle Linux:

```
/var/log/utsetup.year_month_date_hour:minute:second.log
```

Oracle Solaris:

```
/var/adm/log/utsetup.year_month_date_hour:minute:second.log
```

The values in the file name reflect a time stamp of when the commands are started. Check these files for notices of installation problems.

See [Section 3.2.12, “Installation \(utinstall\) Error Messages”](#) for a listing of `utinstall` error messages.

11. Reconfigure the Sun Ray network based on the previous configuration you confirmed in step 4.

For a shared network (LAN) with external DHCP server support (used <code>utadm -L on</code>)	No action is required. This configuration is preserved during the upgrade.
For a shared network (LAN) with Sun Ray server DHCP support	# <code>/opt/SUNWut/sbin/utadm -A subnet</code>
For a private network	# <code>/opt/SUNWut/sbin/utadm -a intf</code>

12. If you used a preserve file as part of the upgrade, you must run `utconfig -w` to complete the upgrade.

The `utconfig -w` command will prompt you for the Admin GUI settings, including the location of the Tomcat installation, and the Admin GUI will be started automatically.

13. If the Windows connector groupname was not previously configured or the groupname was set to `root` or `sys`, reconfigure the Windows connector. Errors regarding these scenarios will be listed in the installation log.

```
# /opt/SUNWut/sbin/utconfig -c
```

14. Repeat Steps 1 through 13 for each server in failover group.

15. Synchronize the updated Sun Ray Operating Software (firmware) on the Sun Ray clients.

You must perform this task on a stand-alone Sun Ray server or the last Sun Ray server upgraded in a failover group. The `utfwsync` takes the currently installed and configured firmware on the Sun Ray server and updates all the Sun Ray servers in the failover group, and then it updates all the firmware on the Sun Ray Clients. The Sun Ray Clients reboot themselves and update to the new firmware if needed.

```
# /opt/SUNWut/sbin/utfwsync
```

16. If you plan to use the Windows connector, upgrade the Windows Connector Components on your designated Windows Server.

There is no upgrade program for the Windows connector components. To upgrade a Windows system with the previous components installed, remove the current Windows connector components and install the new versions.

3.5.3. Planning Upgrades Using Failover Groups

By configuring two or more Sun Ray servers in a failover group, you can reduce interruption of new service availability in the event that one server fails. If you plan to combine existing Sun Ray servers into a failover group, or to upgrade an existing failover group, please consider the following:

- You should always upgrade the secondary servers first before upgrading the primary server. New functionality from the release may not work until all the servers in the failover group are upgraded.
- Before you upgrade a given server, make sure that Sun Ray Client users terminate their sessions.



Note

If upgrading servers in a large configuration at once is not convenient, upgrade one or two servers at a time until the entire configuration is complete.

- For best results in groups of four or more servers, configure the primary server so that it serves only the Sun Ray data store. Configure the secondary servers so that they serve users directly in addition to serving the data store.
- While upgrading the primary server, secondary servers will not be able to do any updates to the data store.
- To take advantage of the new features in this release, do not mix different Sun Ray Software versions within a failover group. Failover groups that use more than one software version revert to the functionality of the earliest version.
- Using the Admin GUI to restart or reset Sun Ray services does not work across servers with different Sun Ray releases. For example, even if you use the Admin GUI to restart all the servers in a failover group that are running the latest Sun Ray Software release, you should still restart or reset any Sun Ray servers running earlier versions of Sun Ray Software.
- Disable all Sun Ray Client firmware updates until all the servers in a failover group are upgraded. For details, see [Section 14.13, “How to Disable All Sun Ray Client Firmware Updates”](#).

**Note**

Even if you upgrade one or two servers per week, you must wait until all servers in the group are upgraded before you enable firmware updates.

- If your configuration is a dedicated private interconnect, disconnect the server from the Sun Ray interconnect.

See [Chapter 6, Failover Groups](#) for a more general discussion of failover groups, including diagrams of failover topologies.

3.5.4. How to Preserve Sun Ray Software Configuration Data

When you choose an upgrade, the `utsetup` script automatically preserves your existing configuration information. You must preserve your existing configuration before running the `utsetup` script only in the following situations:

- Upgrading the operating system on an existing Sun Ray server that requires you to reformat the server's disk.
- Replacing an existing Sun Ray server hardware with a new server.
- Upgrading the operating system on the Sun Ray server, in most situations, as part of the Sun Ray Software upgrade.

In all of these cases, you will need to create a Sun Ray Software configuration data backup file, saved in `/var/tmp/SUNWut.upgrade/preserve_version.tar.gz`, and copy it to the same location on the newly installed or upgraded server before you start the `utsetup` script. The `utsetup` script automatically restores the configuration data in the `/var/tmp/SUNWut.upgrade/preserve_version.tar.gz` after it installs Sun Ray Software.

The `utpreserve` script in the Sun Ray Software image directory preserves the following information:

- X user settings
- Sun Ray data store
- Authentication Manager configuration files
- `utslaunch` properties
- Failover group information
- Kiosk mode configuration
- Group name used by the Windows connector

The `utpreserve` script does **not** preserve the following information:

- The Sun Ray server's network and DHCP configuration settings (`utadm -A` or `utadm -a` configuration information). You must reconfigure those settings after upgrading Sun Ray Software.
- The server's PAM configuration is not saved. The PAM configuration is located in `/etc/pam.conf` on Oracle Solaris 10 and `/etc/pam.d/*` on Oracle Solaris 11 or Oracle Linux. You need to back up and restore the PAM configuration manually.

Before You Begin

Depending on the size of your configuration, this procedure, including the operating system software upgrade, might take anywhere from five minutes to several hours or even more to complete.

**Note**

Running the `utpreserve` script stops all Sun Ray daemons and services, including the Sun Ray data store, causing users to lose all of their sessions, both active and disconnected. Make sure to inform them of your plans.

Steps

1. Change directory to the unzipped Sun Ray Software media pack.
2. Preserve the Sun Ray configuration:

```
# ./utpreserve
```

The `utpreserve` script warns that it will stop all Sun Ray services, consequently terminating all user sessions, and asks whether it should continue.

If you answer **y**, the `utpreserve` script:

- Stops the Sun Ray services and the Sun Ray data store daemon.
- Lists the files that are saved.
- Tars and compresses the entire list of files as the `/var/tmp/SUNWut.upgrade/preserve_version.tar.gz` file, where `version` is the currently installed version of Sun Ray Software.
- Indicates that a log file containing notices of errors is available at `/var/adm/log/utpreserve.year_month_date_hour:minute:second.log` for Oracle Solaris or `/var/log/SUNWut/utpreserve.year_month_date_hour:minute:second.log` for Oracle Linux
where `year`, `month`, and so on are represented by numeric values reflecting the time `utpreserve` was started.
- Recommends that the `/var/tmp/SUNWut.upgrade/preserve_version.tar.gz` backup file be copied to a safe location.

**Note**

If you have modified the PAM configuration in a previous version of Sun Ray Software, your changes might be lost when Sun Ray Software is upgraded. To avoid losing your modifications, be sure to save a copy before performing the update, then use the saved copy to restore your earlier modifications.

Chapter 4. Admin GUI and Commands

Table of Contents

4.1. Sun Ray Software Commands	53
4.1.1. How to Set Up Access to the Sun Ray Software Man Pages	56
4.2. Administration Tool (Admin GUI)	56
4.2.1. Administrative Name and Password	57
4.2.2. Admin GUI Tab Descriptions	57
4.2.3. How to Log In to the Administration Tool (Admin GUI)	59
4.2.4. How to Change the Admin GUI Locale	60
4.2.5. How to Change the Admin GUI to English Locale	60
4.2.6. How to Change the Admin GUI Timeout	60
4.2.7. How to Enable or Disable Multiple Administration Accounts (Oracle Linux)	61
4.2.8. How to Enable or Disable Multiple Administration Accounts (Oracle Solaris 11)	62
4.2.9. How to Enable or Disable Multiple Administration Accounts (Oracle Solaris 10)	62
4.2.10. How to Audit Admin GUI Sessions	63

This chapter provides the list of Sun Ray Software commands and details about the Sun Ray Software Administration GUI, known as the Admin GUI.

4.1. Sun Ray Software Commands

[Table 4.1, “Sun Ray Software Commands”](#) lists the important commands used to administer the Sun Ray environment. For further information, see the man page for the command in question.

The Sun Ray Software commands are located in the following directories:

- `/opt/SUNWut/bin`
- `/opt/SUNWut/sbin`
- `/opt/SUNWuttsc/bin/`
- `/opt/SUNWuttscwrap/bin/` (Oracle Solaris)

Add these directories to your PATH variable, so you don't need to provide the full path when using the commands.

Table 4.1. Sun Ray Software Commands

Command	Definition
<code>utaction</code>	Execute commands when a Sun Ray Client session is connected, disconnected, or terminated.
<code>utadm</code>	Manage the private network, shared network, and DHCP (Dynamic Host Configuration Protocol) configuration for the Sun Ray interconnect.
<code>utadminuser</code>	Add, list, and delete UNIX user names from the list of users authorized to administer Sun Ray services. The list is stored in the Sun Ray data store.
<code>utamghadm</code>	Configure or disable regional hotdesking, which enables users to access their sessions across multiple failover groups.
<code>utaudio</code>	Enable Sun Ray audio services.
<code>utcapture</code>	Monitor packets sent and dropped between the Sun Ray server and the Sun Ray Clients.

Command	Definition
<code>utcard</code>	Enable the configuration of different types of smart cards in the Sun Ray data store.
<code>utconfig</code>	Perform the initial configuration of the Sun Ray server and supporting administration framework software.
<code>utcrypto</code>	Configure the Sun Ray server security.
<code>utdesktop</code>	Manage Sun Ray Clients connected to the Sun Ray server on which the command is run.
<code>utdetach</code>	Disconnect the current non-smart card mobile session or authenticated smart card session from its respective Sun Ray Client. The session is not destroyed but put into a detached state. The session can be accessed again only after authentication. When Remote Hotdesk Authentication (RHA) is disabled (through <code>utpolicy</code> or the Admin GUI), <code>utdetach</code> affects only authenticated smart card sessions and non-smart card mobile sessions.
<code>utdevadm</code>	Enable or disable Sun Ray device services. The services include USB devices connected through USB ports, embedded serial ports, clipboard, internal smart card reader in the Sun Ray Client, and the scbus protocol.
<code>utdiskadm</code>	Manage Sun Ray mass storage.
<code>utdssync</code>	Convert the port number for the Sun Ray data store service to the new default port on servers in a failover group, then forces all servers in the group to restart Sun Ray services.
<code>uteject</code>	Eject media from a removable storage media device.
<code>utfwadm</code>	Manage firmware versions on the Sun Ray Clients.
<code>utfwinstall</code>	Install the Sun Ray Operating Software (Sun Ray Client firmware).
<code>utfwload</code>	Force the download of new firmware to a Sun Ray Client running older firmware than its server.
<code>utfwsync</code>	Refresh the firmware level on the Sun Ray Clients to the level available on the Sun Ray servers in a failover group. It then forces all the Sun Ray Clients within the group to restart.
<code>utfwuninstall</code>	Uninstall the Sun Ray Operating Software (Sun Ray Client firmware).
<code>uthashpwd</code>	Create a hashed version of a password used for authenticating the enabling or disabling of the Configuration GUI.
<code>utgmtarget</code>	Manage a group-wide list of explicit destinations for Sun Ray group membership announcements.
<code>utgroupsig</code>	Set the failover group signature for a group of Sun Ray servers. The <code>utgroupsig</code> command also sets the Sun data store <code>rootpw</code> used by Sun Ray to a value based on the group signature. Although <code>utgroupsig</code> sets the <code>rootpw</code> in the <code>utdsd.conf</code> file, it does <i>not</i> set the admin password, which is a separate entity, in the data store.
<code>utgstatus</code>	View the failover 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.
<code>utinstall</code>	Install, upgrade, and remove the Sun Ray Software.
<code>utkeyadm</code>	Manage Sun Ray Client device keys for authentication.
<code>utkeylock</code>	Modify the state of certain locking modifier keys on a user's keyboard.
<code>utkiosk</code>	Import or export kiosk configuration information into the data store. It also supports storage of multiple named kiosk session configurations in the data store.
<code>utkioskoverride</code>	Set the session type associated with a token, to select a kiosk session configuration for a token associated with a kiosk session, or to query the session type and kiosk session currently associated with a token.

Command	Definition
<code>utlicenseadm</code>	Manage Sun Ray device licenses.
<code>utmhadm</code>	Manage multihead groups using a CLI.
<code>utmhconfig</code>	Manage multihead groups using a GUI.
<code>utmount</code>	Mount a file system on a Sun Ray mass storage device.
<code>utpolicy</code>	Set and report the policy configuration of the Sun Ray Authentication Manager, <code>utauthd</code> .
<code>utpreserve</code>	Save existing Sun Ray Software configuration data to the <code>/var/tmp/SUNWut.upgrade</code> directory.
<code>utpw</code>	Change the Sun Ray administrator password (also known as the UT admin password) used by the Web-based and command-line administration applications.
<code>utquery</code>	Display a Sun Ray Client's current parameter values.
<code>utreader</code>	Add, remove, and configure token readers.
<code>utrelease</code>	Display the version of the Sun Ray Software and the Sun Ray Operating Software installed on the server.
<code>utreplica</code>	Configure the Sun Ray data store server to enable replication of administered data from a designated primary server to each secondary server in a failover group. The data stores of the secondary servers remain synchronized automatically unless there is a power outage.
<code>utresadm</code>	Control the resolution and refresh rate of the video monitor signal (persistent monitor settings) produced by the Sun Ray Client.
<code>utresdef</code>	Create, delete, and view resolution definitions, that is, monitor signal timing definitions for monitors attached to Sun Ray Clients.
<code>utscreenresize</code>	Resize user's screen to its optimal size, which is useful for hotdesking.
<code>utselect</code>	Present the output of <code>utswitch -l</code> as a list of servers in the current host group, to be used for reconnection of the current Sun Ray Client. A user can either select a server from this list or specify a server not in the current host group by typing its full name in the <code>utselect</code> text box.
<code>utsession</code>	List and manage Sun Ray sessions on the local Sun Ray server.
<code>utset</code>	View and change Sun Ray Client settings.
<code>utsettings</code>	Open the Sun Ray Settings GUI to view or change audio and visual settings for the Sun Ray Client.
<code>utsetup</code>	Run all the appropriate commands, including <code>utinstall</code> and <code>utconfig</code> , in the appropriate sequence to install and configure a Sun Ray server.
<code>utstart</code>	Start Sun Ray services.
<code>utstop</code>	Stop Sun Ray services.
<code>utswitch</code>	Switch a Sun Ray Client session to another Sun Ray server. The <code>utswitch</code> command can also list existing sessions for the current token.
<code>uttsc</code>	Start a Windows connector session, providing a remote desktop.
<code>uttscwrap</code>	Start a Windows connector session, providing a remote desktop. This command is an optimized version of the <code>uttsc</code> command when using JDS on a Sun Ray server running Oracle Solaris.
<code>utumount</code>	Unmount a file system from a Sun Ray mass storage device.

Command	Definition
<code>utuser</code>	Report Sun Ray user token registrations and manage those registrations. The <code>utuser</code> command is able to obtain smart card token values from Sun Ray Clients that are configured as dedicated token reader devices.
<code>utwall</code>	Send a message or an audio file to users having an Xnewt or Xsun (X server unique to Sun Ray) process. The messages can be sent in email and displayed in a pop-up window.
<code>utwho</code>	Assemble information about display number, token, logged-in user, and the like, in a compact format.
<code>utxconfig</code>	Manage X server configuration parameters for users of Sun Ray Client sessions.
<code>utxlock</code>	Lock a Sun Ray desktop session.

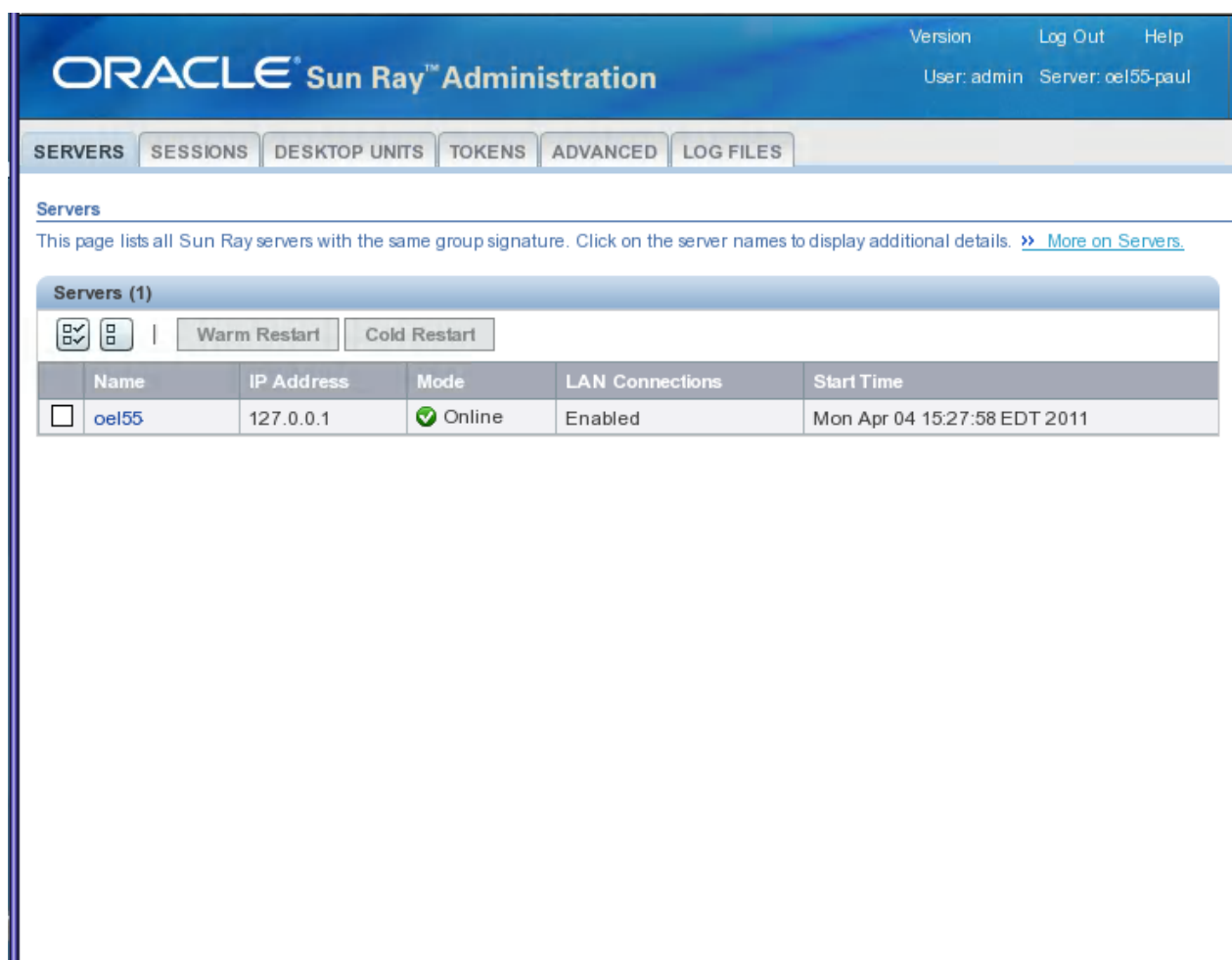
4.1.1. How to Set Up Access to the Sun Ray Software Man Pages

Add the following paths to the `MANPATH` environment variable:

- `/opt/SUNWut/man`
- `/opt/SUNWkio/man`
- `/opt/SUNWuttsc/man`
- `/opt/SUNWuttscwrap/man`

4.2. Administration Tool (Admin GUI)

The Sun Ray Administration Tool (Admin GUI) is organized around primary Sun Ray objects such as servers, sessions, desktop units, and tokens. Each type of object has a dedicated tab that provides related functionality. [Figure 4.1, “Admin GUI Home Screen”](#) shows the home screen.

Figure 4.1. Admin GUI Home Screen

4.2.1. Administrative Name and Password

The default user name for the administration account is `admin`.

The password is set during the Sun Ray server configuration. If you can't remember the administration password, you can use the `utconfig -w` command to reconfigure the administration software, including the password. To change the administration password, use the Advanced tab in the Admin GUI or the `utpw` command.

To allow another user account to perform administrative functions, see [How to Enable or Disable Multiple Administration Accounts \(Oracle Linux\)](#) or [How to Enable or Disable Multiple Administration Accounts \(Oracle Solaris\)](#).

4.2.2. Admin GUI Tab Descriptions

Table 4.2, "Admin GUI Tab Descriptions" describes tabs provided with the Admin GUI. See [Appendix B, Admin GUI Help](#) for detailed reference information.

Table 4.2. Admin GUI Tab Descriptions

Tab	Functions
Servers	From the Servers tab, you can do the following tasks:

Tab	Functions
	<ul style="list-style-type: none"> List all of the servers in the failover group. Display the host group's network connectivity status. Show the host group's installed Sun Ray packages. Display details about each server. Perform a warm restart of Sun Ray services on a local or failover group basis. A warm restart does not terminate sessions prior to the restart. Perform a cold restart of Sun Ray services on a local or failover group basis. A cold restart terminates all sessions on the selected servers prior to the restart.
Sessions	<p>From the Sessions tab, you can do the following tasks:</p> <ul style="list-style-type: none"> List all the sessions, sorted by user sessions and idle sessions. Use the search function to find specific sessions such as those running on a single server or sessions where a specific user is logged in. Select a session's server to display details about the server or client and to select and terminate sessions.
Desktop Units	<p>From the Desktop Units tab, which includes the Sun Ray Clients and Oracle Virtual Desktop Clients, you can do the following tasks:</p> <ul style="list-style-type: none"> List all registered clients. List all connected clients. List all clients configured as token readers. List all clients participating in multihead groups.
Tokens	<p>From the Tokens tab, you can do the following tasks:</p> <ul style="list-style-type: none"> Manage the tokens associated with users. Manage the pseudo-tokens associated with clients.
Advanced	<p>The Advanced tab includes the following subtabs:</p> <p>Security Subtab</p> <p>From the Security subtab, you can disable and re-enable security settings, such as encryption of communication between client and server, server authentication, security mode, and device access.</p> <p>System Policy Subtab</p> <p>From the System Policy subtab, you can regulate authentication manager policy settings, such as:</p> <ul style="list-style-type: none"> Access for card users and non-card users, which includes enabling Kiosk Mode, Oracle Virtual Desktop Client access, or Mobile Sessions. Enabling Client Authentication

Tab	Functions
	<ul style="list-style-type: none"> • Enabling the Multihead feature, • Session Access when Hotdesking
	Kiosk Mode Subtab From the Kiosk Mode subtab, you can configure Kiosk Mode for your system.
	Card Probe Order Subtab From the Card Probe Order subtab, you can rearrange the order that smart cards are probed. You can move the cards that are used most frequently to the top of the list.
	Data Store Password Subtab From the Data Store Password subtab, you can change the password for the administrator account.
Log Files	From the Log Files tab, you can do the following tasks: <ul style="list-style-type: none"> • View Sun Ray system messages. • View authentication events. • View server administration events. • View mount messages. • View storage events.

All actions performed within the Admin GUI that modify system settings are logged in an audit trail.

4.2.3. How to Log In to the Administration Tool (Admin GUI)

This procedure describes how to log in to the Sun Ray Administration tool.



Note

If a session is inactive for 30 minutes, you must log in again. To change the timeout value, see [How to Change the Admin GUI Timeout](#).

1. Log in to your Sun Ray server's console or to any client attached to it.
2. Open a browser window and type the following URL:

```
http://localhost:1660
```



Note

If you specified a different port number when you configured the Sun Ray Software, use that port number in the URL. If you enabled secure communication, the browser might be redirected to a secure port. The default secure port is 1661.

3. In the User Name window, type the administrator user name and click **OK**.
4. In the password challenge screen, type the administration password and click **OK**.

The Sun Ray Administration tool is displayed.

If you get a message denying access, check the following items:

- You are running a browser on a Sun Ray server or one of its clients.
- The browser is not using a different machine as an HTTP proxy server.

If you get a blank browser page:

- To access the Admin GUI from a system instead of the Sun Ray server, you must have remote access enabled (it is disabled by default). To enable remote access to the Admin GUI, unconfigure the Admin GUI using the `utconfig -w -u` command and then run `utconfig -w` to reconfigure. Choose Yes to enable remote access.

4.2.4. How to Change the Admin GUI Locale

To display the locale correctly in the Admin GUI, change your browser's language preferences to the desired locale.

4.2.5. How to Change the Admin GUI to English Locale

This procedure describes how to change the Admin GUI to display English if it is displaying an undesired language.

1. Become superuser on the Sun Ray server.
2. Export the English locale.

```
export LC_ALL=C
```

3. Stop the web admin services.

```
/etc/init.d/utwadmin stop
```

4. Start the web admin services.

```
/etc/init.d/utwadmin start
```

For a more permanent solution, you can remove the non-English Sun Ray Software packages from the server. The following example removes the French packages and restarts the web admin services.

```
# /etc/init.d/utwadmin stop
# pkgrm SUNWfuta SUNWfuta SUNWfutwh SUNWfutwl
# /etc/init.d/utwadmin start
```

4.2.6. How to Change the Admin GUI Timeout

This procedure describes how to change the timeout for the Admin GUI. By default, the Admin GUI timeout value is 30 minutes.

1. Become superuser on the Sun Ray server.
2. Edit the `/etc/opt/SUNWut/webadmin/webadmin.conf` configuration file.
3. Change the following timeout value:

```
...
# The session timeout (specified in minutes)
session.timeout=30
```

...

- Restart the `webadmin` program.

```
# /opt/SUNWut/lib/utwebadmin restart
```

This tool automatically updates the `web.xml` file used by the web server hosting the Admin GUI.

4.2.7. How to Enable or Disable Multiple Administration Accounts (Oracle Linux)

The Sun Ray server administrator can allow any valid UNIX user ID, which has been added to the `utadmin` authorized user list, to administer Sun Ray services using the Admin GUI. An audit trail of activity on these accounts is provided. The `utadminuser` command enables you to add existing UNIX users to the `utadmin` authorized user list.

Authentication for accounts with administrative privileges is based on the PAM authentication framework.

4.2.7.1. How to Configure Admin GUI Privileges for UNIX Users (Oracle Linux)

Use the following procedure to configure the Sun Ray Admin GUI to allow access by the UNIX users in the `utadmin` authorized user list instead of the default `admin` account. Once you enable Admin GUI privileges for authorized users, you can add or remove users to the `utadmin` authorized list to manage access to the Admin GUI.

- For each UNIX user that needs authorization to the Admin GUI, add the user to the authorized user list.

```
# utadminuser -a username
```

You can run the `utadminuser` command without any options to list the current authorized users or with the `-d` option to delete a user.

- Add the following auth entries to the `/etc/pam.d/utadmingui` file:

```
##PAM-1.0
# BEGIN: added to utadmingui by SunRay Server Software -- utadmingui
auth include system-auth
# END: added to utadmingui by SunRay Server Software -- utadmingui
```



Note

Make sure to include the comment lines, which are needed for the cleanup to work properly.

4.2.7.2. How to Limit Admin GUI Privileges to the Admin User (Oracle Linux)

This is the default Admin GUI privilege configuration when the Sun Ray Software is installed.

To limit Admin GUI privileges to the default `admin` user, replace the PAM entries in the `/etc/pam.d/utadmingui` file with the `pam_sunray_admingui.so.1` module.

```
# BEGIN: added to utadmingui by SunRay Server Software -- utadmingui
auth sufficient /opt/SUNWut/lib/pam_sunray_admingui.so.1
# END: added to utadmingui by SunRay Server Software -- utadmingui
```



Note

Make sure to include the comment lines, which are needed for the cleanup to work properly.

4.2.8. How to Enable or Disable Multiple Administration Accounts (Oracle Solaris 11)

The Sun Ray server administrator can allow any valid UNIX user ID, which has been added to the `utadmin` authorized user list, to administer Sun Ray services using the Admin GUI. An audit trail of activity on these accounts is provided. The `utadminuser` command enables you to add existing UNIX users to the `utadmin` authorized user list.

Authentication for accounts with administrative privileges is based on the PAM authentication framework.

4.2.8.1. How to Configure Admin GUI Privileges for UNIX Users (Oracle Solaris 11)

Use the following procedure to configure the Sun Ray Admin GUI to allow access by the UNIX users in the `utadmin` authorized user list instead of the default `admin` account. Once you enable Admin GUI privileges for authorized users, you can add or remove users to the `utadmin` authorized list to manage access to the Admin GUI.

1. For each UNIX user that needs authorization to the Admin GUI, add the user to the authorized user list.

```
# utadminuser -a username
```

You can run the `utadminuser` command without any options to list the current authorized users or with the `-d` option to delete a user.

2. Add the following auth entries to the `/etc/pam.d/utadmingui` file:

```
##PAM-1.0
# BEGIN: added to utadmingui by SunRay Server Software -- utadmingui
auth requisite pam_authtok_get.so.1
auth required pam_dhkeys.so.1
auth required pam_unix_cred.so.1
auth required pam_unix_auth.so.1
```



Note

Make sure to include the comment lines, which are needed for the cleanup to work properly.

4.2.8.2. How to Limit Admin GUI Privileges to the Admin User (Oracle Solaris 11)

This is the default Admin GUI privilege configuration when the Sun Ray Software is installed.

To limit Admin GUI privileges to the default `admin` user, replace the PAM entries in the `/etc/pam.d/utadmingui` file with the `pam_sunray_admingui.so.1` module.

```
##PAM-1.0 # BEGIN: added to utadmingui by SunRay Server Software -- utadmingui
auth sufficient /opt/SUNWut/lib/pam_sunray_admingui.so.1
```



Note

Make sure to include the comment lines, which are needed for the cleanup to work properly.

4.2.9. How to Enable or Disable Multiple Administration Accounts (Oracle Solaris 10)

The Sun Ray server administrator can allow any valid UNIX user ID which has been added to the `utadmin` authorized user list to administer Sun Ray services using the Admin GUI. An audit trail of activity on

these accounts is provided. The `utadminuser` command enables you to add existing UNIX users to the `utadmin` authorized user list.

Authentication for accounts with administrative privileges is based on the PAM authentication framework.

4.2.9.1. How to Configure Admin GUI Privileges for UNIX Users (Oracle Solaris 10)

Use the following procedure to configure the Sun Ray Admin GUI to allow access by the UNIX users in the `utadmin` authorized user list instead of the default `admin` account. Once you enable Admin GUI privileges for authorized users, you can add or remove users to the `utadmin` authorized list to manage access to the Admin GUI.

1. For each UNIX user that needs authorization to the Admin GUI, add the user to the authorized user list.

```
# utadminuser -a username
```

You can run the `utadminuser` command without any options to list the current authorized users or with the `-d` option to delete a user.

2. Modify the `/etc/pam.conf` file to use the `other` authentication PAM stack auth entries to create the PAM stack for `utadmingui`

```
# BEGIN: added to utadmingui by SunRay Server Software -- utadmingui
utadmingui auth requisite pam_authtok_get.so.1
utadmingui auth required pam_dhkeys.so.1
utadmingui auth required pam_unix_cred.so.1
utadmingui auth required pam_unix_auth.so.1
```



Note

Make sure to include the comment line, which is needed for the cleanup to work properly.

4.2.9.2. How to Limit Admin GUI Privileges to the Admin User (Oracle Solaris 10)

This is the default Admin GUI privilege configuration when the Sun Ray Software is installed.

To limit Admin GUI privileges to the default `admin` user, modify the `/etc/pam.conf` file and replace the PAM stack for `utadmingui` with the `pam_sunray_admingui.so.1` module.

```
# BEGIN: added to utadmingui by SunRay Server Software -- utadmingui
utadmingui auth sufficient /opt/SUNWut/lib/pam_sunray_admingui.so.1
```



Note

Make sure to include the comment line, which is needed for the cleanup to work properly.

4.2.10. How to Audit Admin GUI Sessions

The administration framework provides an audit trail of the Admin GUI. The audit trail is an audit log of the activities performed by multiple administration accounts. All events that modify system settings are logged in the audit trail. Sun Ray Software uses the `syslog` implementation.

The events are logged in the following log file:

```
/var/opt/SUNWut/log/messages
```

All audit events are prefixed with the keyword `utadt::` so you can filter events from the `messages` file.

For example, session termination from the Admin GUI generates the following audit event:

```
Jun 6 18:49:51 sunrayserver usersession[17421]: [ID 521130 user.info] utadt:: username= /  
{demo} hostname={sunrayserver} service={Sessions}  
cmd={/opt/SUNWut/lib/utrcmd sunrayserver /opt/SUNWut/sbin/utsession -x -d 4 -t /  
Cyberflex_Access_FullCrypto.1047750ble0e -k 2>&P1}  
message={terminated User "Cyberflex_Access_FullCrypto.1047750ble0e" with display number="4" on /  
"sunrayserver"}  
status={0} return_val={0}
```

where:

- `username` = User's UNIX ID
- `hostname` = Host on which the command is executed
- `service` = Name of the service being executed
- `cmd` = Name of the command being executed
- `message` = Details about the action being performed

Chapter 5. Sun Ray Server and Networking

Table of Contents

5.1. Log Files	65
5.2. How to Start or Stop Sun Ray Services	67
5.2.1. How to Stop Sun Ray Services	67
5.2.2. How to Start Sun Ray Services (Warm Restart)	67
5.2.3. How to Start Sun Ray Services (Cold Restart)	67
5.3. How to Check and Fix Corrupted Configuration Files (Oracle Solaris 10)	68
5.4. How to Unconfigure a Sun Ray Server	69
5.5. How to Disconnect a Sun Ray Server From the Interconnect	69
5.6. User Fields in the Sun Ray Data Store	70
5.7. Network Troubleshooting	70
5.7.1. Network Load	70
5.7.2. The <code>utcapture</code> Utility	70
5.7.3. <code>utcapture</code> Examples	71
5.7.4. The <code>utquery</code> Command	71

This chapter provides information about administering the Sun Ray server, including log files and managing Sun Ray services. Network troubleshooting information is also provided.

5.1. Log Files

Significant activity occurring on the Sun Ray server is logged and saved. The server stores this information in text files.

The structure and content of the various messages written to these files and other Sun Ray Software log files is arbitrary and might change at any time. The messages do not provide a stable interface for programmatic consumption.

Table 5.1, “Sun Ray Server Log Files” describes the log files that are maintained on the Sun Ray server.

Table 5.1. Sun Ray Server Log Files

Log File	Path	Description
Administration	<code>/var/opt/SUNWut/log/admin_log</code>	Lists operations performed during server administration. This log is updated daily. Archived files are stored on the system for up to one week and are annotated using numeric extensions, for example, from file name <code>admin_log.0</code> to <code>admin_log.5</code> .
Authentication	<code>/var/opt/SUNWut/log/auth_log</code>	Lists events logged from the Authentication Manager. The <code>auth_log</code> file is updated up to a limit of 10 every time the server's authentication policy is changed or started. The archived authentication files are annotated using numeric extensions, for example, from <code>auth_log.0</code> to <code>auth_log.9</code> .

Log File	Path	Description
Automatic mounting	<code>/var/opt/SUNWut/log/utmountd.log</code>	Lists mount messages for mass storage devices. The archived <code>utmountd</code> files are annotated using numeric extensions, for example, from <code>utmountd.log.0</code> to <code>utmountd.log.9</code> .
Mass storage devices	<code>/var/opt/SUNWut/log/utstoraged.log</code>	Lists mass storage device events. The archived storage files are annotated using numeric extensions, for example, from <code>utstoraged.log.0</code> to <code>utstoraged.log.9</code> .
Messages	<code>/var/opt/SUNWut/log/messages</code>	<p>Lists events from the server's clients, including details of registering, inserting, or removing smart cards. This file is updated daily. Archived files are stored up to seven days or 3.5 MB, annotated with numeric extensions, for example, from <code>messages.0</code> to <code>messages.5</code>.</p> <p>When using a shared network (LAN) with external DHCP server support (configured network using <code>utadm -L on</code>), logging for each event type is disabled unless the <code>LogXXX</code> value and the <code>LogHost</code> value are set. See Table 13.4, “Sun Ray Client Configuration Parameters (.parms)” for details.</p>
Web administration	<code>/var/opt/SUNWut/log/utwebadmin.log</code>	Lists web administration-related messages. The archived log files are annotated with numeric extensions.
USB redirection	<code>/var/opt/SUNWut/log/uttsrpd.log</code>	Lists USB redirection messages from the Windows connector.

[Table 5.2, “Sun Ray Server Installation and Configuration Log Files”](#) list the specific installation and configuration logs files.

Table 5.2. Sun Ray Server Installation and Configuration Log Files

Log File	Path	Description
Activation	<ul style="list-style-type: none"> Oracle Solaris - <code>/var/adm/log/utctl.*</code> Oracle Linux - <code>/var/log/SUNWut/utctl.*</code> 	List events logged during post-installation product activation. The archived log files are annotated with time stamp extensions.
Installation	<ul style="list-style-type: none"> Oracle Solaris - <code>/var/adm/log/utinstall.*</code> Oracle Linux - <code>/var/log/utinstall.*</code> 	List events logged during installation. The archived log files are annotated with time stamp extensions.
Setup	<ul style="list-style-type: none"> Oracle Solaris - <code>/var/adm/log/utsetup.*</code> Oracle Linux - <code>/var/log/SUNWut/utsetup.*</code> 	List events logged during installation/configuration when performed by <code>utsetup</code> . The archived log files are annotated with time stamp extensions.

5.2. How to Start or Stop Sun Ray Services

The following procedures describe how to start and stop the Sun Ray services. There are many situations when Sun Ray services need to be restarted, such as when you change one of the configuration parameters in the Sun Ray data store.

5.2.1. How to Stop Sun Ray Services

1. Become superuser on the Sun Ray server.
2. Stop the Sun Ray services.

```
# /opt/SUNWut/sbin/utstop
```

Once stopped, the Sun Ray services will not restart even after the server is rebooted. You must use the `utstart` command to start Sun Ray services.

5.2.2. How to Start Sun Ray Services (Warm Restart)

This procedure, known as a warm restart, starts Sun Ray services without clearing existing sessions.



Note

A disconnect will occur for a brief time on active Sun Ray Clients before they reconnect again.

1. Become superuser on the Sun Ray server.
2. Start the Sun Ray services.

```
# /opt/SUNWut/sbin/utstart
```



Note

The following warning may display: `WARNING: The current DBus file descriptor limit can only support up to approximately 120 logged-in desktops. Please reboot the system if you require support for more desktops.` As instructed, reboot the server if the Sun Ray server is supporting more than 120 desktops.

5.2.3. How to Start Sun Ray Services (Cold Restart)

This procedure, known as a cold restart, starts Sun Ray services and clears existing sessions.



Note

Be sure to notify your users before performing a cold restart, which terminates all existing sessions on a server. To restart Sun Ray services without terminating sessions, perform a warm restart.

1. Become superuser on the Sun Ray server.
2. Start the Sun Ray services.

```
# /opt/SUNWut/sbin/utstart -c
```

**Note**

The following warning may display: `WARNING: The current DBus file descriptor limit can only support up to approximately 120 logged-in desktops. Please reboot the system if you require support for more desktops.` As instructed, reboot the server if the Sun Ray server is supporting more than 120 desktops.

5.3. How to Check and Fix Corrupted Configuration Files (Oracle Solaris 10)

If the `dtlogin` daemon cannot start the `Xsun` or `Xnewt` server properly, the following configuration files might be corrupted:

- `/etc/dt/config/Xservers`
- `/etc/dt/config/Xconfig`

The following procedure explains how to correct this problem.

**Note**

This procedure shows output from a simplified example. Your output may have tens of lines between the `BEGIN SUNRAY CONFIGURATION` and `END SUNRAY CONFIGURATION` comments.

1. As a user of the Sun Ray server, open a shell window and compare the `/usr/dt/config/Xservers` and `/etc/dt/config/Xservers` files.

```
% diff /usr/dt/config/Xservers /etc/dt/config/Xservers
```

This command compares a known good file with the suspect file. The output should be similar to the following example.

```
106a107,130
> # BEGIN SUNRAY CONFIGURATION
> :3 SunRay local@none /etc/opt/SUNWut/basedir/lib/utxsun :3 -nobanner
.
.
> :18 SunRay local@none /etc/opt/SUNWut/basedir/lib/utxsun :18 -nobanner
> # END SUNRAY CONFIGURATION
```

The first line of output contains 106a107,130. The 106 means that the two files are identical to the 106th line of the files. The a107,130 means that the information on lines 107 through 130 of the second file would have to be added to the first file to make it the same as the second file.

If your output shows the first three digits to be a number less than 100, the `/etc/dt/config/Xservers` file is corrupt.

2. Compare the `/usr/dt/config/Xconfig` and `/etc/dt/config/Xconfig` files.

```
% diff /usr/dt/config/Xconfig /etc/dt/config/Xconfig
```

The output should be similar to the following example.

```
156a157,180
> # BEGIN SUNRAY CONFIGURATION
> Dtlogin.*_8.environment: SUN_SUNRAY_TOKEN=ZeroAdmin.ml.at88sc1608.6d0400aa
```

```
.
.
> Dtlogin.*_9.environment: SUN_SUNRAY_TOKEN=ZeroAdmin.ml.at88sc1608.a10100aa
> # END SUNRAY CONFIGURATION
```

If your output shows the first three digits to be a number less than 154, the `/etc/dt/config/Xconfig` file is corrupt.

3. If either file is corrupted, continue this procedure to replace the configuration files.
4. Become superuser on the Sun Ray server and shut down the Sun Ray Client services.



Note

Replacing the `Xservers` file requires shutting down all Sun Ray Client services. Remember to inform users of the outage.

```
# /opt/SUNWut/sbin/utstop
```

5. Replace the `Xservers` and `Xconfig` files as appropriate.

```
# /bin/cp -p /usr/dt/config/Xservers /etc/dt/config/Xservers
# /bin/cp -p /usr/dt/config/Xconfig /etc/dt/config/Xconfig
```



Note

For headless servers, comment out or remove the `:0` entry from the `Xservers` file.

6. Re-initialize the authentication policy.

```
# /opt/SUNWut/sbin/utstart -c
```

The extra lines within the previous `Xservers` and `Xconfig` files are automatically rebuilt.

5.4. How to Unconfigure a Sun Ray Server

1. Become superuser on the Sun Ray server.
2. Remove the replication configuration.

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

3. Unconfigure Sun Ray Software.

```
# /opt/SUNWut/sbin/utconfig -u
```

4. Answer `y` to all the prompts.

5.5. How to Disconnect a Sun Ray Server From the Interconnect



Note

This procedure disconnects users from their sessions on the Sun Ray server. Make sure users terminate their sessions before you continue.

1. Become superuser on the Sun Ray server.
2. Disconnect the Sun Ray server from the Sun Ray interconnect.

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

**Note**

(Oracle Solaris Only) If you press **Ctrl-C** while performing **utadm** configuration, the Admin GUI may not function correctly the next time you invoke it. To correct this condition, use the **dhtadm -R** command.

5.6. User Fields in the Sun Ray Data Store

The Sun Ray data store is a private data store service used for access to Sun Ray Software administration and configuration data. The data store is useful for maintaining consistency across failover groups.

Table 5.3, “Sun Ray Data Store User Fields” describes the user fields in the Sun Ray data store.

Table 5.3. Sun Ray Data Store User Fields

Fields	Description
Token ID	User's unique token type and ID. For smart cards, this value is a manufacturer type and the card's serial ID. For clients, this value is the type "pseudo" and the client's Ethernet address. Examples: <code>mondex.9998007668077709</code> <code>pseudo.080020861234</code>
Server Name	Name of the Sun Ray server that the user is using. This setting is optional.
Server Port	Sun Ray server's communication port. This field should generally be set to 7007. This setting is optional.
User Name	User's name.
Other Info	Any additional information you want to associate with the user, for example, an employee or department number. This setting is optional.

5.7. Network Troubleshooting

This section describes some potential reasons for slow network performance and the tools to help troubleshoot network problems.

To help you tune the network performance in a Sun Ray environment, see [Chapter 20, Performance Tuning](#).

5.7.1. Network Load

In situations where network load or packet loss is too high, network cables or switch equipment might be defective in very rare cases.

1. Verify that network connections are 100F.
2. Use **utcapture** to assess network latency and packet loss.

As latency and packet loss increase, performance suffers.

5.7.2. The **utcapture** Utility

The **utcapture** utility connects to the Sun Ray Authentication Manager and reports packet loss statistics and round-trip latency timings for each client connected to this server. See the **utcapture** man page to learn more about this command.

Table 5.4, “[utcapture Output](#)” describes the information that [utcapture](#) outputs.

Table 5.4. [utcapture](#) Output

Data Element	Description
TERMINALID	The MAC address of the client.
TIMESTAMP	The time the loss occurred in year-month-day-hour-minute-second format, for example, 20041229112512.
TOTAL PACKET	Total number of packets sent from the server to the client.
TOTAL LOSS	Total number of packets reported as lost by the client.
BYTES SENT	Total number of bytes sent from the server to the client.
PERCENT LOSS	Percentage of packets lost between the current and previous polling interval.
LATENCY	Time in milliseconds for a round trip from the client to the server.

5.7.3. [utcapture](#) Examples

The following command captures data every 15 seconds from the Authentication Manager running on the local host and then writes it to stdout if any change occurs in packet loss for a client.

```
% utcapture -h |
```

The following command captures data every 15 seconds from the Authentication Manager running on the local host and then writes it to stdout.

```
% utcapture -r > raw.out
```

The following command captures data every 15 seconds from the Authentication Manager running on server5118.eng and then writes the output to stdout if any change occurs in packet loss for the client with ID 080020a893cb or 080020b34231.

```
% utcapture -s sunray_server5118.eng 080020a893cb 080020b34231
```

The following command processes the raw data from the input file [raw-out.txt](#) and then writes to stdout the data only for those clients that had packet loss.

```
% utcapture -i raw-out.txt
```

5.7.4. The [utquery](#) Command

The [utquery](#) command interrogates a client and displays the client's initialization parameters with the IP addresses of the DHCP services that supplied those parameters. This command can be helpful in determining whether a client was able to obtain the parameters that were expected in a particular deployment and in determining specific DHCP servers that contributed to the clients initialization. See the [utquery](#) man page to learn more about this command.

Chapter 6. Failover Groups

Table of Contents

6.1. Failover Groups Overview	73
6.2. Failover Process	74
6.3. Load Balancing	74
6.4. Mixing Different Sun Ray Servers	74
6.5. Authentication Requirements	75
6.6. Dedicated Primary Servers for Data Store	75
6.7. Setting Up a Failover Group	75
6.7.1. How to Configure a Primary Server	76
6.7.2. How to Add a Secondary Server	76
6.7.3. How to Synchronize Primary and Secondary Sun Ray Servers	77
6.7.4. How to Change the Group Manager Signature	77
6.8. Additional Failover Group Tasks	78
6.8.1. How to Take a Server Offline and Online	78
6.8.2. How to Disable Load Balancing	78
6.8.3. How to Show the Current Sun Ray Data Store Replication Configuration	78
6.8.4. How to Remove the Replication Configuration	78
6.8.5. How to View the Failover Group Status	78
6.9. Recovery Issues and Procedures	79
6.9.1. How to Rebuild the Primary Server's Administration Data Store	79
6.9.2. How to Replace the Primary Server with a Secondary Server	80
6.9.3. Secondary Server Recovery	80
6.10. Group Manager Details	80
6.10.1. Group Manager Configuration	81

This chapter describes how to setup and manage a failover group.

6.1. Failover Groups Overview

A failover group consists of two or more servers that provide users with a high level of availability in case one server becomes unavailable. When a server in a failover group becomes unavailable, whether for maintenance, a power outage, or any other reason, each desktop client connected to it reconnects to another server in the failover group that has an existing session for the client's current token. If the client can't find an existing session for the current token, it connects to a server selected by the load balancing algorithm. This server presents a login screen to the user, who then logs in to create a new session. The session on the failed server is lost.

A failover group consists of a primary server and one or more secondary servers, configured with the `utreplica` command. Each Sun Ray server hosts its own local Sun Ray data store. However, only read access is permitted on the local data stores. Any data changes (write access) are first written on the primary server and later replicated on the secondary servers' Sun Ray data stores.

Servers in a failover group authenticate (or learn to trust) one another by using a common group signature, a key used to sign messages sent between servers in the group. The group signature must be configured, using the `utgroupsig` command, to be identical on each server. See [Section 6.7.4, “How to Change the Group Manager Signature”](#) for details. The `admin` password asked during `utconfig` also needs to be identical on all the Sun Ray servers in a failover group. See [Section 4.2.1, “Administrative Name and Password”](#) for details.

You can use the `utgstatus` command or the Servers tab in the Admin GUI to see the status of all servers in a failover group. The following server modes affect how the server participates in a failover group:

- Online: The server participates in the normal session creation process controlled by the load balancing algorithm in the failover group.
- Offline: The server does not participate in load balancing any more (the load balancing algorithm does not select this server for new sessions), although sessions can still be created on it, either explicitly, through the use of the `utswitch` or `utselect` command, or implicitly, if all other servers are down.

When using a failover group, clients are automatically redirected to servers based on load balancing before sessions are created. However, users can use the `utselect` or `utswitch` commands for manual redirection.

The `utselect` GUI is the preferred method to use for server selection. For more information, see the `utselect` man page.

For more information about how to set up multiple failover groups that use regional hotdesking, see [Chapter 9, Hotdesking](#).

6.2. Failover Process

For the Sun Ray Software, the Group Manager manages the failover process. For every server in a failover group, the Group Manager does the following actions:

- Detects the presence of other Sun Ray servers belonging to the same group.
- Monitors the availability (liveness) of the other servers.
- Exchanges information about allocation of sessions and server load for load balancing purposes.
- Facilitates the redirection of clients to other servers when needed.

6.3. Load Balancing

When a server in a failover group fails, the Group Manager on each remaining server distributes the failed server's clients to create new sessions among the remaining servers.

When the Group Manager receives a token from a Sun Ray Client for which no server owns an existing session, it redirects the client. This redirection is determined according to the result of a load-sensitive session placement lottery conducted among the servers in the group, based on each server's capacity (number and speed of its CPUs), load, number of sessions, and other factors.



Note

Load balancing is handled automatically, as described. The administrator may choose to turn load balancing off but cannot assign values or otherwise modify the algorithm.

6.4. Mixing Different Sun Ray Servers

There may be situations when you need to mix different Sun Ray servers in a failover group, either based on the hardware architecture, operating system, or Sun Ray Software version. The following list describes which mix of Sun Ray servers are supported in a failover group.

- Mixing SPARC-based and x86-based servers running Oracle Solaris in the same failover group is supported.

- Mixing servers running Oracle Solaris and Oracle Linux in the same failover group is not supported.
- Mixing servers running different versions of Sun Ray Software is supported. However, this is not recommended for long-term use in a production environment. Failover groups that use more than one version of Sun Ray Software will be unable to use all the features provided in the latest releases.

**Note**

When multiple versions of Sun Ray Software are used in a failover group, the primary server should run the oldest of the versions in use. Otherwise, the presence of a newer feature on the primary server might prevent proper replication of the Sun Ray data store to secondary servers that are running older versions.

- If a Sun Ray environment has Unix desktop users, then all the Sun Ray servers in the failover group should run the same version of the operating system. Otherwise, using different GNOME desktop versions from the same home directory can lead to a corrupted desktop state.

6.5. Authentication Requirements

All of the servers in a failover group should use the same authentication mechanism and name service. For example, a failover group should not use one server with NIS authentication and other servers with NIS+ authentication.

If a failover group does have different authentication mechanisms, the servers might fail to obtain their existing non-smart card mobile (NSCM) sessions if a user name contains the same characters but the case is different, for example, `ps121664`, `PS121664`, or `Ps121664`.

6.6. Dedicated Primary Servers for Data Store

When configuring failover groups, the Sun Ray data store is automatically configured to enable replication of the Sun Ray administration data across the primary and secondary Sun Ray servers. In large failover groups, significant loads may be pushed onto the primary server from various sources. In addition, runaway processes from user applications on the primary can degrade the health of the entire failover group.

Failover groups of more than four servers should have a dedicated primary server solely devoted to serve the Sun Ray data store. For example, a dedicated primary server should not host any Sun Ray sessions. To configure the primary server so that it serves only the Sun Ray data store, use the `utadm -f` command to take the server offline. Secondary servers should always serve user sessions and the data store.

6.7. Setting Up a Failover Group

Use the following procedures to set up a Failover Group.

1. Set up server addresses and client addresses. This step is needed only when using a private network or a when using a Sun Ray server that provides DHCP services. See [Chapter 19, Alternate Network Configurations](#) for details.
2. Configure a primary server.
3. Add secondary servers.
4. Synchronize the primary and secondary servers.
5. Change the group manager signature.

6.7.1. How to Configure a Primary Server

Layered administration of the failover group takes place on the primary server, where the master copy of the Sun Ray data store resides. The `utreplica` command designates a primary server, advises the server of its administration primary status, and tells it the host names of all the secondary servers.

The term primary server reflects the replication relationship, not the failover order.

Before You Begin

- Configure the primary server before you add the secondary servers.
- If a common home directory is mounted on machines with different GNOME versions, conflicts between or among the versions cause unpredictable behavior. Do not try to use multiple GNOME versions with a common home directory.

Steps

1. Become superuser on the primary Sun Ray server.
2. Configure this server as the primary Sun Ray server and identify all secondary servers.

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

where *secondary_server1* [*secondary_server2...*] is a space-separated list of unique host names of the secondary servers.

When the script ends, a log file is available at:

For Oracle Solaris:

```
/var/adm/log/utreplica.Year_Month_Date_Hour:Minute:Second.log
```

For Oracle Linux:

```
/var/log/SUNWut/utreplica.Year_Month_Date_Hour:Minute:Second.log
```

6.7.2. How to Add a Secondary Server

The secondary servers in the group store a replicated version of the primary server's administration data.

Use the `utreplica` command to advise each secondary server of its secondary status and also the host name of the primary server for the group. Adding or removing secondary servers requires services to be restarted on the primary server.

1. If the secondary server has not been configured on the primary server, become superuser on the primary server and rerun the `utreplica` command with the new secondary server.

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

where *secondary_server1* [*secondary_server2...*] is a space-separated list of unique host names of the secondary servers.

2. Become superuser on the secondary server.
3. Add the secondary server.

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

where `primary-server` is the host name of the primary server.

6.7.3. How to Synchronize Primary and Secondary Sun Ray Servers

Log files for Sun Ray servers contain time-stamped error messages that can be difficult to interpret if the time is out of sync. To make troubleshooting easier, make sure that all secondary servers periodically synchronize with their primary server.

The Network Time Protocol (NTP) is the recommended protocol to synchronize primary and secondary servers. With NTP, you can synchronize to an absolute time source and it provides additional synchronization capabilities. In some deployments, the simpler TIME protocol configured through the `rdate` command may be sufficient.



Note

Both the NTP and TIME protocols are disabled by default on Oracle Solaris servers.

6.7.4. How to Change the Group Manager Signature

The `utconfig` command asks for a group manager signature if you chose to configure for failover. The signature, which is stored in the `/etc/opt/SUNWut/gmSignature` file, must be the same on all servers in the group.

The location can be changed in the `gmSignatureFile` property of the `auth.props` file.

To form a fully functional failover group, the signature file must meet the following criteria:

- Owned by root with only root permissions
- Contain at least eight characters, in which at least two characters are letters and at least one character is not



Note

For slightly better security, use long passwords.

Steps

1. Become superuser on the Sun Ray server.
2. Start the `utgroupsig` command.

```
# /opt/SUNWut/sbin/utgroupsig
```

You are prompted for the signature.
3. Enter the signature twice identically for acceptance.
4. For each Sun Ray server in the group, repeat this procedure.



Note

Be sure to use the `utgroupsig` command rather than any other method to provide the signature. `utgroupsig` also ensures proper internal replication.

6.8. Additional Failover Group Tasks

6.8.1. How to Take a Server Offline and Online

Being able to take servers offline makes maintenance easier. In an offline state, no new sessions are created. However, old sessions continue to exist and can be reactivated unless the Sun Ray Software is affected.

- Take a server offline:

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

- Take a server online:

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

6.8.2. How to Disable Load Balancing

In the `auth.props` file, set `enableLoadBalancing` to `false`.

6.8.3. How to Show the Current Sun Ray Data Store Replication Configuration

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

The result indicates whether the server is stand-alone, primary (with the secondary host names), or secondary (with the primary host name).

6.8.4. How to Remove the Replication Configuration

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

6.8.5. How to View the Failover Group Status



Note

Sun Ray server broadcasts do not traverse routers or servers other than Sun Ray servers.

Command-Line Steps

- View the failover group status for the local Sun Ray server:

```
# /opt/SUNWut/sbin/utgstatus
```

Admin GUI Steps

1. Click the Servers tab.
2. Select a server name to display its Server Details screen.
3. Click **View Network Status**.

The Network Status screen is displayed.

The Network Status screen provides information on group membership and network connectivity for trusted servers, which are those servers in the same failover group.

6.9. Recovery Issues and Procedures

If one of the servers of a failover group fails, the remaining group members operate from the administration data that existed prior to the failure. The recovery procedure depends on the severity of the failure and whether a primary or secondary server has failed.



Note

When the primary server fails, you cannot make administrative changes to the system. For replication to work, all changes must succeed on the primary server.

6.9.1. How to Rebuild the Primary Server's Administration Data Store

Use this procedure to rebuild the primary server's data store from a secondary server, and is performed on the server that was the primary server after it is fully operational again. This procedure uses the same host name for the replacement server.



Caution

Be sure to set `umask` appropriately before running `utldbmc`. Otherwise, unprivileged users can gain access to the `utadmin` password.

1. On one of the secondary servers, capture the current data store to a file called `/tmp/store`

```
# /opt/SUNWut/srds/lib/utldbmc /var/opt/SUNWut/srds/dbm.ut/id2entry.dbb > /tmp/store
```

This command provides an LDIF format file of the current data store.

2. Use FTP to send this file to the `/tmp` directory on the primary server.
3. Follow the Sun Ray Software installation instructions.
4. After running `utinstall`, configure the server as a primary server for the group.

Make sure that you use the same admin password and group signature.

```
# utconfig
:
# utreplica -p secondary-server1 [secondary-server2...]
```

5. Shut down the Sun Ray services, including the data store.

```
# /opt/SUNWut/sbin/utstop
```

6. Restore the data.

```
# /opt/SUNWut/srds/lib/utldif2ldb -c -j 10 -i /tmp/store
```

This command populates the primary server and synchronizes its data with the secondary server. The replacement server is now ready for operation as the primary server.

7. Restart Sun Ray services.

```
# /opt/SUNWut/sbin/utstart -c
```

8. (Optional) Confirm that the data store is repopulated.

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

9. (Optional) Perform any additional configuration procedures.

6.9.2. How to Replace the Primary Server with a Secondary Server



Note

This procedure is also known as promoting a secondary server to primary.

1. Choose a server in the existing failover group to be promoted and configure it as the primary server.

```
# utreplica -u  
# utreplica -p secondary-server1 [secondary-server2...]
```

2. Reconfigure each of the remaining secondary servers in the failover group to use the new primary server:

```
# utreplica -u  
# utreplica -s new-primary-server
```

This command resynchronizes the secondary server with the new primary server.



Note

This process may take some time to complete, depending on the size of the data store. Since Sun Ray services will be offline during this procedure, you may want to schedule your secondary servers' downtime accordingly. Be sure to perform this procedure on each secondary server in the failover group.

6.9.3. Secondary Server Recovery

If a secondary server fails, administration of the group can continue. A log of updates is maintained and applied automatically to the secondary server once it has recovered. If the secondary server needs to be reinstalled, repeat the steps described in the Sun Ray Software installation instructions.

6.10. Group Manager Details

Every server has a group manager module that monitors availability and facilitates redirection. It is coupled with the Authentication Manager.

In setting policies, the Authentication Manager uses the selected authentication modules and decides what tokens are valid and which users have access.



Caution

The same policy must exist on every server in the failover group or undesirable results might occur.

The Group Managers create maps of the failover group topology by exchanging *keepalive* messages among themselves. These *keepalive* messages are sent to a UDP port (typically 7009) on all of the configured network interfaces. The *keepalive* message contains enough information for each Sun Ray server to construct a list of servers and the common subnets that each server can access. In addition, the Group Manager tracks the last time that a *keepalive* message was received from each server on each interface.

The *keepalive* message contains the following information about the server:

- Server's host name
- Server's primary IP address

- Elapsed time since the server was booted
- IP information for every interface the server can reach
- Machine information, such as the number and speed of CPUs, configured RAM, and so on
- Load information, such as the CPU and memory utilization, number of sessions, and so on

The last two items are used to facilitate load balancing.

The information maintained by the Group Manager is used primarily for server selection when a token is presented. The server and subnet information is used to determine the servers to which a given client can connect. These servers are queried about sessions belonging to the token. Servers whose last `keepalive` message is older than the timeout are deleted from the list, because either the network connection or the server is probably down.

6.10.1. Group Manager Configuration

The Authentication Manager configuration file, `/etc/opt/SUNWut/auth.props`, contains properties used by the Group Manager at runtime. The properties are:

- `gmport`
- `gmKeepAliveInterval`
- `enableGroupManager`
- `enableLoadBalancing`
- `enableMulticast`
- `multicastTTL`
- `gmSignatureFile`
- `gmDebug`
- `gmTarget`



Note

These properties have default values that are rarely changed. Only Oracle support personnel should direct you to change these values to help tune or debug your systems. Any properties that are changed must be changed for all servers in the failover group because the `auth.props` file must be the same on all servers in a failover group.

Property changes do not take effect until the Authentication Manager is restarted, which you can do by performing a warm restart of the Sun Ray services.

Chapter 7. Sessions and Tokens

Table of Contents

7.1. Sessions Overview	83
7.1.1. Authentication Manager	84
7.1.2. Session Manager	85
7.2. Managing Sessions	86
7.2.1. How to Redirect a Session	86
7.2.2. How to Disconnect a Session	87
7.2.3. How to Terminate a Session	88
7.2.4. How To Identify a Hung Session	88
7.2.5. How To Kill a Hung Session	88
7.3. Tokens	88
7.3.1. Registering Tokens	89
7.3.2. How to Register a Token	89
7.3.3. How to Register a Pseudo-Token	89
7.3.4. How to Enable, Disable, or Delete a Token	90
7.4. Token Readers	90
7.4.1. How to Configure a Token Reader	90
7.4.2. How to Locate a Token Reader	91
7.4.3. How to Get a Token ID From a Token Reader	91
7.5. Session Troubleshooting	92
7.5.1. Problem: The <code>dtlogin</code> daemon cannot start the Xsun or Xnewt server properly.	92

This chapter provides details on desktop sessions and tokens used to manage the sessions.

7.1. Sessions Overview

A desktop session is a group of services or applications associated with an authentication token and controlled by the Session Manager. Desktop sessions reside on a Sun Ray server and can be directed to any Sun Ray Client or Oracle Virtual Desktop Client. From a user's perspective, a desktop session is usually an instance of an OS desktop.

Sessions can be in one of two states, Connected or Disconnected:

- **Connected:** Every session with a connected status is displayed on client. The session is automatically disconnected when the user removes the smart card or explicitly switches the client to a different session, for example, with the `utswitch` or `utselect` commands.
- **Disconnected:** These sessions are still executed on a server but are not connected to a client and, consequently, are not displayed. However, a user can reconnect to a disconnected session, for example by inserting a smart card containing the appropriate token into the card reader on a client. This changes the session's state to connected and causes it to be displayed on that client.

Sessions can further be grouped into two categories:

- **Idle sessions:** These sessions typically display only a login screen (or login greeter such as `dtlogin`) where no user has been logged in yet. The lifetime of these sessions is controlled by the Sun Ray system. For example, disconnected idle sessions are automatically terminated (reaped off) by the system after a specific time interval.
- **User sessions (or non-idle sessions):** These are sessions with UNIX users logged in. Users may start up additional applications from within the sessions, thus potentially consuming a lot of system resources.

User sessions are therefore of more interest to administrators than idle sessions. To free system resources, monitor the number of long-running disconnected user sessions and, when appropriate, terminate sessions that are no longer in use.

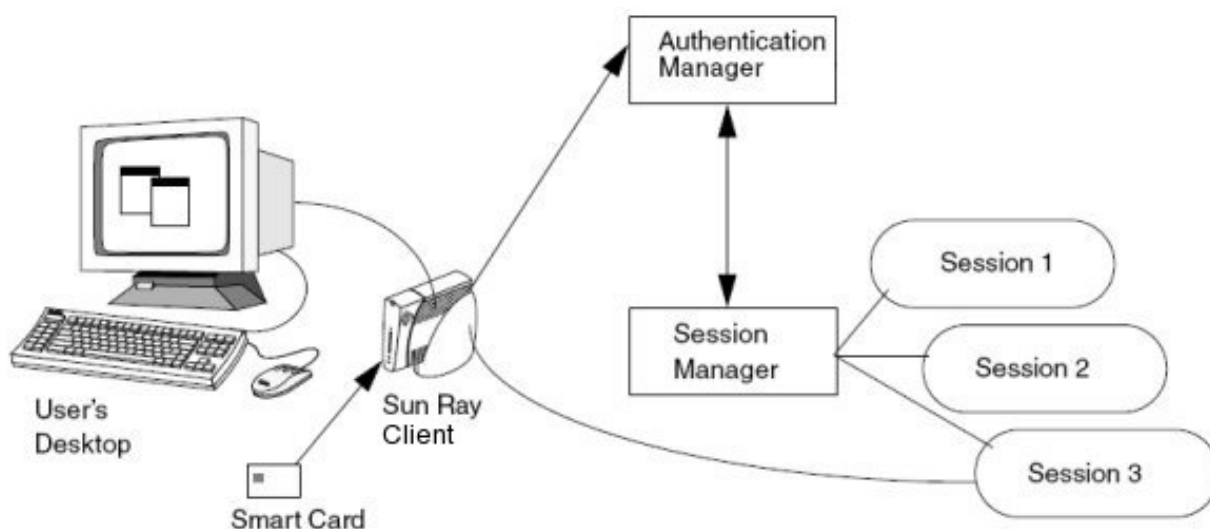
7.1.1. Authentication Manager

The Authentication Manager implements chosen *policies* for identifying and authenticating users on the desktop clients using pluggable components called *modules* to verify user identities, and implements site access policies defined by the administrator. It also supplies an audit trail of the actions of users who have been granted administrative privileges for Sun Ray services. The Authentication Manager is not visible to users.

The interaction between the Authentication Manager and the client is shown in [Figure 7.1, “Authentication and Session Manager Interaction”](#) and works as follows:

1. A user accesses a client.
2. The client sends the user's token information to the Authentication Manager and requests access. If the user inserts a smart card in the client, the card's type and ID are used as the token. If not, the client's Ethernet address is used as a pseudo-token.
3. Based on the policy defined by the system administration, the Authentication Manager accepts or denies the access request.
4. If the user's access request is accepted, the Authentication Manager tells the Session Manager to start an X Windows session, which displays the login screen. Oracle Solaris 10 implementations use the `dtlogin` display manager. Oracle Linux and Oracle Solaris 11 implementations use the `gdm` display manager, or GNOME Display Manager (GDM).

Figure 7.1. Authentication and Session Manager Interaction



If no Sun Ray server is found to provide a session (see [Chapter 2, Planning a Sun Ray Network Environment](#)), the client sends a broadcast request for any Authentication Manager on its subnet.

7.1.2. Session Manager

The Session Manager interacts with the Authentication Manager and directs services to the user. A service is any application that can connect directly to the Sun Ray Client. Such applications can include audio, video, Xservers, and device control of the client. For example, a typical mail application is not a service because it is accessed through an Xserver rather than directly.

The Session Manager is used at startup for services, for managing screen space, and as a rendezvous for the Authentication Manager.

The Session Manager keeps track of sessions and services by mapping services to sessions and binding and unbinding related services to or from a specific client. The Session Manager takes authentication only from authorized Authentication Managers listed in the `/etc/opt/SUNWut/auth.permit` file.

The following sequence describes how the process starts, ends, and restarts:

1. After a user's token is authenticated, the Authentication Manager determines whether a session exists for that token. If a session does not exist, the Authentication Manager asks the Session Manager to create a session and then starts the appropriate services for the session according to the authentication policy decisions taken by the administrator. Creating a session usually requires starting an Xserver process for the session.
2. When services are started, they join the session explicitly by contacting the Session Manager.
3. The Authentication Manager informs the Session Manager that the session associated with the token is to be connected to a specific desktop client. The Session Manager then informs each service in the session that it must connect directly to the client.
4. The user can then interact with the session. The Session Manager mediates control of the screen space between competing services in a session and notifies the services of changes in screen space allocation.
5. When the user removes the smart card, or presses Shift-Pause in an NSCM session, or power cycles the client, or is inactive for longer than the screen lock idle timeout interval, the Authentication Manager determines that the session associated with that token must be disconnected from that client. The Authentication Manager notifies the Session Manager, which in turn notifies all the services in the session and any USB devices to disconnect.
6. When the user re-inserts the smart card, or logs in again to get access to an NSCM session, the Authentication Manager requests the Session Manager to create a new temporary session and then uses it to authenticate the user. This is known as Remote Hotdesk Authentication (RHA). After the user has been authenticated, the desktop client is connected directly to the user's session.



Note

RHA does not apply to anonymous Kiosk Mode or to token readers. Sun Ray Software can be configured to turn this security policy feature off.

The Session Manager is consulted only if the state of the session changes or if other services are added. When a user's token is no longer mapped to a client, for example, when a card is removed, the Session Manager disconnects the services from the client, but the services remain active on the server. For example, programs attached to the Xserver continue to run although their output is not visible. The Session Manager daemon must continue running. To verify that the Session Manager daemon is running, use the `ps` command and look for `utsessiond`.

If the Authentication Manager quits, the Session Manager disconnects all the authorized sessions and requires them to be reauthenticated. These services are disconnected but still active. If the Session

Manager is disrupted, it restarts automatically. Each service contacts the Session Manager to request reattachment to a particular session.

7.2. Managing Sessions

This section the list the tasks to manage sessions on a Sun Ray server.

7.2.1. How to Redirect a Session

A desktop session is automatically redirected to the appropriate Sun Ray server based on the following situations:

- Failover group redirection occurs after token insertion.
- Regional Hotdesking redirection (if configured) occurs after token or user identification and before user authentication.

To manually redirect a session to a different server, use the `utselect` GUI or the `utswitch` command.

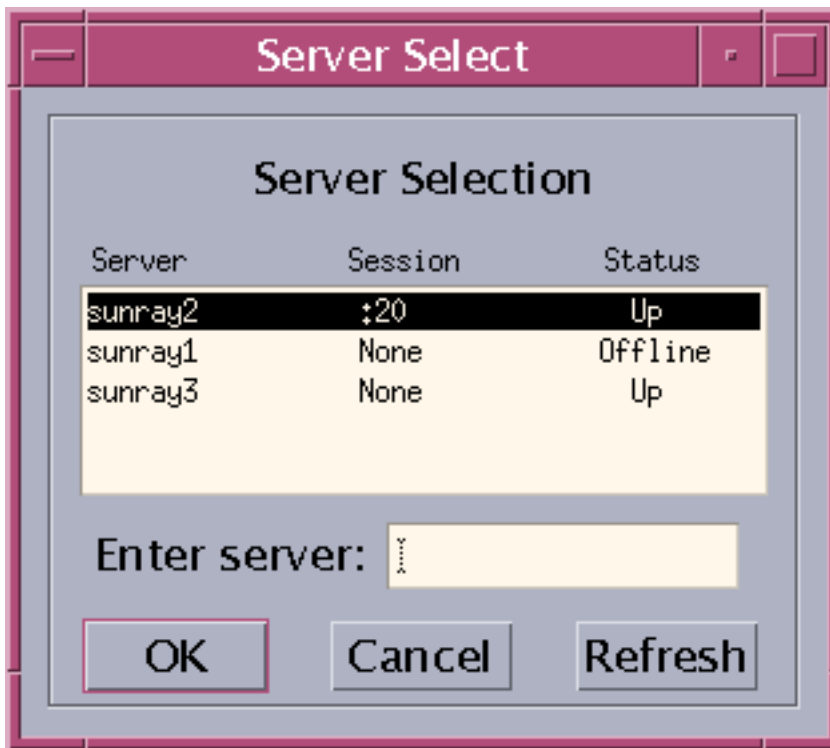
7.2.1.1. How to Manually Redirect to a Different Sun Ray Server (utselect)

- From a shell window on the desktop client, type the following command:

```
% utselect
```

The selections in the window are sorted in order of the most current to least current active sessions for the token ID.

As shown in [Figure 7.2, "Server Selection \(utselect\) GUI"](#), the Server column lists the servers accessible from the desktop client. The Session column reports the DISPLAY variable X session number on the server if one exists. In the Status column, Up indicates that the server is available. The first server in the list is selected by default. Select a server from the list or type the name of a server in the Enter server field. If a server without an existing session is selected, a new session is created on that server.

Figure 7.2. Server Selection (utselect) GUI

7.2.1.2. How to Manually Redirect to a Different Sun Ray Server (utswitch)

- In a shell window on the desktop client, type the following command:

```
% utswitch -h host
```

where *host* is the host name or IP address of the Sun Ray server to which the selected client is redirected.

7.2.2. How to Disconnect a Session

When you disconnect a desktop session, the client stops displaying the user's desktop and requires authentication to re-access the session again in most situations. Disconnecting a session is important for security reasons when you have to leave a client unattended.

When using smart cards, the session disconnects when you remove the smart card from the client. For NSCM and RHA sessions, you can disconnect a client session through any of the following methods:

- (Oracle Solaris only) Lock the desktop through the current desktop manager. For example, in the Java Desktop System, choose **Launch > Lock Screen**. Locking the desktop forces the session to disconnect. A disconnect will also occur if the desktop screen lock idle time interval is exceeded.

Locking an Oracle Linux desktop locks only the desktop and does not disconnect the session. If a new user wants needs to use the desktop client where the desktop is locked, the user must reset the client to disconnect the session and make the client available for use. So, for Oracle Linux desktops, using the `utdetach` command is recommended.

- Use the `utdetach` command:

```
% /opt/SUNWut/bin/utdetach
```

- Press Shift-Pause.

To change the disconnect hot key combination, see [Section 13.2.2, “Sun Ray Client Hot Keys”](#).

**Note**

The hot key combination does not work with a full-screen Windows session.

- Connect to your session through another client, either by inserting your smart card and authenticating to RHA or by logging in through NSCM.

7.2.3. How to Terminate a Session

To terminate the current session and the current X server process, perform one of the following actions:

- Log out from your current desktop session.
- Press the key combination Ctrl-Alt-Bksp-Bksp.

A momentary delay might occur before the session terminates.

**Note**

Use Ctrl-Alt-Bksp-Bksp only for emergencies when you are unable to log out from the desktop. When using this method, applications will not have the opportunity to exit properly and save data, and some application data corruption might result.

7.2.4. How To Identify a Hung Session

1. Become superuser on the Sun Ray server.
2. Type the following command:

```
# /opt/SUNWut/sbin/utdesktop -l -w
```

7.2.5. How To Kill a Hung Session

1. Become superuser
2. Type the following command:

```
# /opt/SUNWut/sbin/utsession -k -t token
```

7.3. Tokens

As described earlier, the Authentication Manager implements the chosen policies for identifying and authenticating users on Sun Ray Clients. Tokens are the key piece for this process.

Sun Ray tokens are authentication keys used to associate a session with a user. A token is a string that consist of a token type and an identifier. If a user inserts a smart card into a client, the card's type and identifier are used as the token (for example `mondex.9998007668077709`). If the user is not using a smart card, the token type `pseudo` and the client's identifier (MAC address) are supplied as the token (for example `pseudo.080020861234`).

The initial token is used to check access rights and to determine the user's session. During this process, the token is eventually translated into other token types (such as escape token, auth token, etc.) used

internally by the Sun Ray system. As an administrator, you rarely need to deal with these internal token types, focusing instead on the initial tokens provided on smart cards or as pseudo-tokens.

7.3.1. Registering Tokens

You can also register smart card tokens and pseudo-tokens in the Sun Ray data store to assign them to specific users (also known as token owners).

Although it requires more setup and administration to register smart cards, there a number of reasons to register tokens:

- Storing the owner's name as well as any other information that helps you to manage tokens in your organization.
- Creating alias tokens to enable users to access the same session with multiple tokens. For example, if a user loses a smart card, you can register a new smart card as a replacement. This will be an alias token.
- Overriding the group-wide kiosk mode setting specified on the System Policy page. If kiosk mode functionality has been configured on your system, you can also specify, for each token, whether the user should be directed to a regular (non-kiosk) session or a Kiosk session when the token is inserted.

Much like the `utuser` command, the Tokens tab in the Admin GUI lists all tokens currently registered in the Sun Ray data store. You can search for specific tokens by entering a search string that includes parts of either the token identifier, owner, or other information. The Search menu enables you to limit the scope of the search further, so that it is also possible to display all currently used tokens, regardless of their registration.

The Policy tab (under the Advanced tab) makes it possible to define high-level access rules for either smart card access or pseudo-token access as well as access rights for registered tokens (see the Policy help page).

You can administer tokens through the `utuser` command or the Admin GUI.

7.3.2. How to Register a Token

This procedure describes how to register a token using the Admin GUI.

1. Click the Tokens tab.
2. Select a token to display that token's properties.
3. Click **New**.
4. Type an identifier or select a token reader.

7.3.3. How to Register a Pseudo-Token

This procedure describes how to register a pseudo-token with the Admin GUI.

1. Click the Desktop Units tab.
2. Select any Desktop Unit Identifier to view properties for that client.
3. On the Desktop Unit Properties page, click **View Token Details**.
4. Click **Edit** to display the Edit Token Properties page.

5. Provide details such as ownership and to specify a session type: Default, Kiosk, or Regular.

7.3.4. How to Enable, Disable, or Delete a Token

This procedure describes how to enable, disable, or delete a token with the Admin GUI.

1. Select the token's identifier on the Token Properties page.
2. Click **Enable**, **Disable**, or **Delete**.

7.4. Token Readers

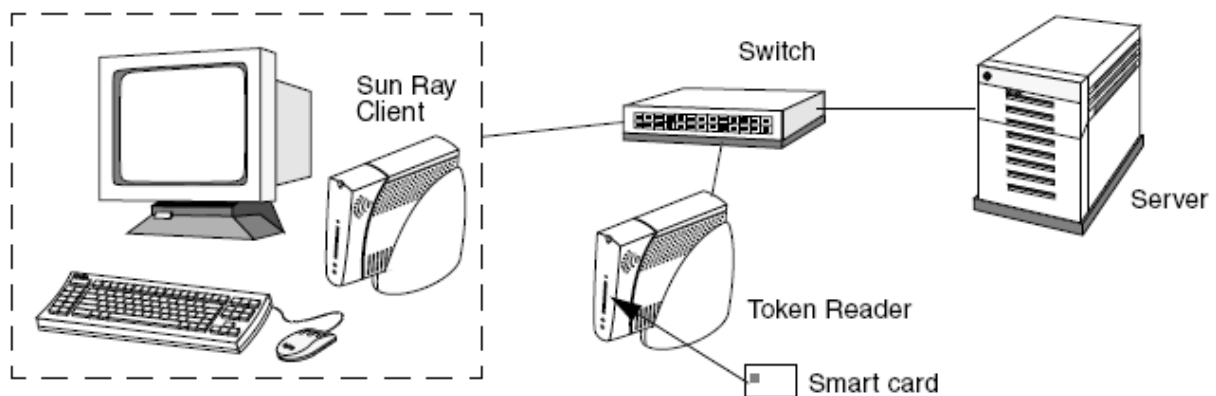
A token reader is a specific Sun Ray Client that you can set up to administer user's tokens, such as registering smart cards. This token reader is not the same as hardware devices into which users insert their smart cards, which are typically called smart card readers. Site administrators can use these token readers to administer Sun Ray tokens, such as assigning a token to a user (token owner).

Sun Ray Software provides a way to designate one or more specific Sun Ray Clients as dedicated token readers. A dedicated token reader is not used for normal Sun Ray services, so it does not need a keyboard, mouse, or monitor. Inserting a smart card in a token reader does not enable hotdesking. It does allow the administrator to assign the card to a user.

When you enable an authentication policy with registered users or token owners, be sure to specify smart card IDs for them. To use token readers with regional hotdesking based on Sun Ray pseudo-tokens, use the Site-specific Mapping Library.

Figure 7.3, "Token Reader Setup" shows that the second client is used as a token reader.

Figure 7.3. Token Reader Setup



7.4.1. How to Configure a Token Reader

Command Line Steps

The `utreader` command enables a client to be used as a token reader for registering smart cards. When a client is configured as a token reader, inserting or removing a smart card does not initiate session mobility. Any session connected to that client remains connected to it regardless of card movement events.

Token reader mode is useful when you want to determine the raw token ID of a smart card.

- To configure the Sun Ray Client with MAC address `0800204c121c` as a token reader:


```
# utreader -a 0800204c121c
```

- To re-enable the Sun Ray Client with MAC address `0800204c121c` to recognize card movement events and perform session mobility based on the smart card inserted into the client:

```
# utreader -d 0800204c121c
```

- To unconfigure all token readers on this server:

```
# utreader -c
```

Admin GUI Steps

1. Click the Desktop Units tab.
2. Click the identifier of the client that you want to use as a token reader.
3. On the Desktop Units Properties window, click **Edit**.
4. On the Edit Desktop Unit Properties window, select the **Token Reader** option.
5. Click **OK**.

The client you have selected is now set up to read smart card tokens.

6. Restart Sun Ray services.

The client is now a token reader.

7.4.2. How to Locate a Token Reader

This procedure describes how to locate a token reader using the Admin GUI.

1. Click the Desktop Units tab.
2. Select **Token Readers** from the drop-down list.
3. Click **Search**.

The default search finds all possible matches.

To change the search criteria, type text in the Search text box.

7.4.3. How to Get a Token ID From a Token Reader

You can access the token card reader by invoking `utuser -r` from any server in the relevant failover group.

Type the following command:

```
# utuser -r token-reader
```

where `token-reader` is the MAC address of the client containing the smart card whose ID you want to read. Insert the smart card into the client and run the `utuser` command. This command queries the client for the smart card token's ID and, if successful, displays it. For example:

```
# /opt/SUNWut/sbin/utuser -r 08002086e18f  
Insert token into token reader '08002086e18f' and press return.
```

```
Read token ID 'mondex.9998007668077709'
```

7.5. Session Troubleshooting

The Sun Ray administration model has seven types of user sessions:

- Default - Normal user login
- Register - User self-registration
- Kiosk - Anonymous user operation
- Insert card - User smart card required
- Card error - Unrecognized user smart card type
- No entry - User's smart card token is blocked
- Session Refused - The server refuses to grant a session to a client that does not meet the server's security requirements

The Default, Register, and Kiosk session types have normal login processes. When a problem occurs, investigate the following:

- Sun Ray server configuration files. However, the Sun Ray Software itself modifies some configuration files. In most cases, these changes are identified with Sun Ray Software specific comments. Do not change these modifications.
- Any X server startup files that have been modified
- `dtlogin` or `gdm` status

Although the last four session types display icons on the Sun Ray Client, they do not have login processes. If the user removes and reinserts the smart card immediately, the icon disappears but the Wait for Session OSD icon remains. These session types and their OSDs do not cause concern. The user can perform one of the following actions:

- Insert a recognized smart card in the correct orientation
- Ask the Sun Ray administrator to grant access
- Ask the Sun Ray administrator to download the correct firmware

7.5.1. Problem: The `dtlogin` daemon cannot start the Xsun or Xnewt server properly.

See [Section 5.3, "How to Check and Fix Corrupted Configuration Files \(Oracle Solaris 10\)"](#).

Chapter 8. Smart Card Services

Table of Contents

8.1. Overview	93
8.2. Smart Card Bus Protocol	94
8.3. Smart Card Configuration Files	94
8.4. Smart Card Probe Order	94
8.5. Hotdesking with Smart Cards	95
8.6. Configuring Smart Card Services	95
8.6.1. How to Configure Primary Smart Card Readers for Hotdesking and Authentication	95
8.6.2. How to Configure External CCID-Compliant USB Smart Card Readers for Authentication (Oracle Solaris)	96
8.6.3. How to Add a Smart Card Configuration File	96
8.6.4. How to Change the Smart Card Probe Order	97
8.6.5. How to Change the Smart Card Bus Protocol (Oracle Solaris)	97
8.7. Troubleshooting Smart Card Services	98
8.7.1. Smart Card Transaction Problems	98
8.8. CCID IFD Handler for External USB Smart Card Readers (Oracle Solaris)	99
8.8.1. How to Install CCID IFD Handler	99
8.8.2. How to Uninstall CCID IFD Handler	99
8.8.3. Known Issues	100

This chapter provides details about the smart card services provided by Sun Ray Software.

8.1. Overview

Sun Ray Software automatically provides smart card services, such as smart card authentication, through the PC/SC-lite API. PC/SC (Personal Computer/Smart Card) is the standard framework for smart card device access on multiple OS platforms.

Smart card services include interoperability with the integrated smart card reader on a Sun Ray Client or an enabled smart card reader connected to a client computer running Oracle Virtual Desktop Client. For Sun Ray servers running Oracle Solaris, external CCID-compliant USB smart card readers on both Sun Ray Clients and Oracle Virtual Desktop Clients are supported with the CCID IFD handler, which can be downloaded separately.

Custom applications are frequently used to provide the following solutions:

- Strong smart card-based authenticated logins and PKCS#11
- S/MIME digital signature message signing and encryption

All Sun Ray 2 Series Clients and Sun Ray 3 Series Clients support smart cards that operate at all three of the ISO-7816 defined Vcc voltages of 1.8 Volts (Class C), 3 Volts (Class B), and 5 Volts (Class A). The client firmware automatically chooses the most appropriate voltage to operate the card. There are no administrator settings available or required to control this feature.

For the latest list of smart cards tested with Sun Ray Software, see the [Smart Cards for Hotdesking](#) page on Oracle Technology Network.

8.2. Smart Card Bus Protocol

Sun Ray Software uses a customized smart card bus protocol, called scbus, to exchange smart card transactions between a Sun Ray server and its desktop clients. Previous to Sun Ray Software 5.3, there was only one version of the scbus protocol, which is referred to as scbus v1. With Sun Ray Software 5.4, scbus v2 is available and it provides the following enhancements:

- Extended APDU support (ISO-7816-4 2005 rev)
- Compliance with the PC/SC 2.0 IFD Handler API
- Smart card services for Oracle Virtual Desktop Clients
- Protocol and parameters selection (PPS) for Sun Ray 3 Series Clients

The scbus v2 protocol is supported on all Sun Ray 2 Series Clients, Sun Ray 3 Series Clients, and Oracle Virtual Desktop Client version 3.1 or later. Sun Ray 1 Series Clients do not support the scbus v2 protocol. If the scbus v2 protocol is enabled on a Sun Ray server, Sun Ray 1 Series Clients will not be able to use smart card services either natively on the default desktop or through the Windows connector.

By default, scbus v1 is enabled on Sun Ray servers running Oracle Solaris and scbus v2 is enabled on Sun Ray servers running Oracle Linux. The scbus v1 protocol is not supported on Sun Ray servers running Oracle Linux. If you need to change the smart card bus protocol on Sun Ray servers running Oracle Solaris, see [Section 8.6.5, “How to Change the Smart Card Bus Protocol \(Oracle Solaris\)”](#).

8.3. Smart Card Configuration Files

When using a smart card on a desktop client, the Sun Ray server must identify the smart card type being used and extract a unique identifier from the card. The smart card types are identified through smart card configuration files, which are written in the SwapDrop language and located in the: `/etc/opt/SUNWut/smartcard` directory. All files end with a `.cfg` suffix, as in, `acme_card.cfg`.

When a user inserts a smart card into a desktop client, the available smart card configuration files are run in order based on the smart card probe order, as described in [Section 8.4, “Smart Card Probe Order”](#). When run, the configuration file performs an I/O action with the smart card to try to identify it.

If a configuration file identifies the smart card, a unique token ID is sent to the server and the session associated with that token ID becomes the active session connected to the client. If no configuration files can identify the card, then the [Card Error icon](#) is displayed on the client and there is no session activated for the client.

8.4. Smart Card Probe Order

The order in which smart cards are inspected is called the probe order. Every time a smart card is inserted into a desktop client, the Sun Ray server tries to identify the card type using the specified probe order. The `utcard` command or the **Advanced > Card Probe Order** page in the Admin GUI enables you to set up a group-wide probe order, which is stored in the Sun Ray data store. Only smart cards identified by one of the configuration files specified in the probe order list are accepted. You can add or remove smart card configuration files from this list to restrict session access to specific card types.

In the absence of a group-wide probe order, the Sun Ray server uses the local probe order defined in the `/etc/opt/SUNWut/smartcard/probe_order.conf` file. If no local probe order has been set up, a default probe order is used. Changes in smart card probe order require Sun Ray services to be restarted.

8.5. Hotdesking with Smart Cards

Unless you use the non-smart card mobility (NSCM) feature, smart cards must be used to hotdesk from one client to another. For Sun Ray Clients, only the internal smart card reader can be used for hotdesking. External smart card readers for hotdesking are not supported.

For Oracle Virtual Desktop Clients, any connected smart card reader can be used for hotdesking. To configure a smart card reader for hotdesking, you need to use the Smart Card setting to enable smart card access and to choose the smart card reader if there is more than one connected. Once configured, the smart card reader uses the scbus channel and is handled just like the internal smart card reader on a Sun Ray Client. See the [Oracle Virtual Desktop Client User's Guide](#) for details.

8.6. Configuring Smart Card Services

This section provides the various procedures needed to configure smart card services on a Sun Ray server, so users can use smart cards on their desktop clients for hotdesking and authentication.

8.6.1. How to Configure Primary Smart Card Readers for Hotdesking and Authentication

This procedure describes how to configure smart card services for the primary smart card reader on a desktop client, which is the internal smart card reader on a Sun Ray Client or the configured smart card reader connected to a client computer running Oracle Virtual Desktop Client. To configure a smart card reader on Oracle Virtual Desktop Client, you need to use the Smart Card setting to enable smart card access and to choose the smart card reader if there is more than one connected. Once configured, the smart card reader uses the scbus channel and is handled just like the internal smart card reader on a Sun Ray Client.

1. Become superuser on the Sun Ray server.
2. Enable smart card services for the primary smart card reader.

By default, smart card services for the primary smart card reader is enabled. You can use the `utdevadm` command to display which services are enabled or disabled. If smart card services are disabled, use the following steps to enable it.

```
# /opt/SUNWut/sbin/utdevadm -e -s internal_smartcard_reader
```



Note

You can also use the Admin GUI to enable smart card services. Select the **Internal Smart Card Reader** option on the **Advanced > Security** page.

3. (Oracle Solaris only) To gain the latest smart card services functionality, including support for Oracle Virtual Desktop Clients, make sure the scbus v2 protocol is enabled.

The scbus v1 protocol is enabled by default. See [Section 8.6.5, "How to Change the Smart Card Bus Protocol \(Oracle Solaris\)"](#) for details.

4. Restart Sun Ray services.

```
# /opt/SUNWut/sbin/utstart -c
```

5. (Windows connector) To enable the smart card reader for the Windows desktop, use the `-r scard:on` of the `uttsc` command.

See [Section 17.13, "Smart Cards"](#) for more details.

8.6.2. How to Configure External CCID-Compliant USB Smart Card Readers for Authentication (Oracle Solaris)

This procedure describes how to configure smart card services for external CCID-compliant USB smart card readers connected to a Sun Ray Client or a client computer running Oracle Virtual Desktop Client. CCID-compliant USB smart card readers are redirected through the Windows RDP smart card channel, which enables the smart card to be used for Windows session authentication.

This procedure applies only to Sun Ray servers running Oracle Solaris, because the CCID IFD handler software is not supported on Sun Ray servers running Oracle Linux.



Note

You can use external USB smart card readers that are not CCID-compliant, but they will be redirected to the Windows desktop through USB redirection. Because USB redirection is available after the user logs in, those smart card readers cannot be used for Windows authentication.

1. Become superuser on the Sun Ray server.
2. Enable USB devices on the Sun Ray server.

By default, external USB device services are enabled. You can use the `utdevadm` command to display which services are enabled or disabled. If USB devices disabled, use the following steps to enable it.

```
# /opt/SUNWut/sbin/utdevadm -e -s usb
```



Note

You can also use the Admin GUI to enable smart card services. Select the **USB Port** option on the **Advanced > Security** page.

3. To gain the latest smart card services functionality, including support for Oracle Virtual Desktop Clients, make sure the scbus v2 protocol is enabled.

The scbus v1 protocol is enabled by default. See [Section 8.6.5, “How to Change the Smart Card Bus Protocol \(Oracle Solaris\)”](#) for details.

4. Install the CCID IFD handler software.

See [Section 8.8, “CCID IFD Handler for External USB Smart Card Readers \(Oracle Solaris\)”](#) for installation and troubleshooting information.

5. Restart Sun Ray services.

```
# /opt/SUNWut/sbin/utstart -c
```

6. (Windows connector) To enable the smart card reader for the Windows desktop, use the `-r scard:on` of the `uttsc` command.

See [Section 17.13, “Smart Cards”](#) for the additional steps to configure smart card services on a Windows system.

8.6.3. How to Add a Smart Card Configuration File

This procedure describes how to add a smart card configuration file to the Sun Ray data store. Once added, the configuration file is automatically assigned the last position in the smart card probe order.

1. Become superuser on the Sun Ray server.
2. Copy the smart card configuration file to the `/etc/opt/SUNWut/smartcard` directory.

The file name must end with a `.cfg` suffix.

3. Add the smart card configuration file.

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

4. Restart the Sun Ray services.

```
# /opt/SUNWut/sbin/utstart -c
```

5. Verify that the card was added.

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

8.6.4. How to Change the Smart Card Probe Order

Admin GUI Steps

1. Click the Advanced tab.
2. Click the Card Probe Order subtab.
3. Change the order of the smart cards.
4. Click **Set Probe Order**.
5. Click **Cold Restart** on the Servers page to restart Sun Ray services.

Command Line Steps

1. Become superuser on the Sun Ray server.
2. List the current order of the smart cards.

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

3. Change the probe order of a smart card.

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

4. Restart the Sun Ray services for the new order to take effect.

```
# /opt/SUNWut/sbin/utstart -c
```

8.6.5. How to Change the Smart Card Bus Protocol (Oracle Solaris)



Note

The scbus v2 protocol is the default protocol for Oracle Linux and it cannot be changed.

By default, scbus v1 is enabled on a Sun Ray server running Oracle Solaris. Choose the scbus version based on your environment:

- scbus v1 - Specify if managing Sun Ray Clients running Sun Ray Software 5.2 firmware or earlier.

- scbus v2 - Specify if managing Sun Ray Clients running Sun Ray Operating Software 11.0 or later or if managing Oracle Virtual Desktop Clients version 3.1 or later.

Admin GUI Steps

1. Click the Advanced tab.
2. Click the Security subtab.
3. In the Devices section, choose the scbus version in the **Internal Smart Card Reader** field under the Devices section.
4. Click **Save**.
5. Click **Cold Restart** on the Servers page to restart Sun Ray services.

Command Line Steps

1. Become superuser on the Sun Ray server.
2. Change the smart card bus protocol.

```
# /opt/SUNWut/sbin/utdevadm -p scbus -v version-number
```

where *version-number* can be *v1* or *v2*. You can use the `-p scbus` with no other options to view the current scbus version set on the Sun Ray server.

3. Restart the Sun Ray services for the new protocol to take effect.

```
# /opt/SUNWut/sbin/utstart -c
```

8.7. Troubleshooting Smart Card Services

Here are some basic checks to make when troubleshooting smart card services.

- Are the smart card readers configured? See [Section 8.6.1, “How to Configure Primary Smart Card Readers for Hotdesking and Authentication”](#) or [Section 8.6.2, “How to Configure External CCID-Compliant USB Smart Card Readers for Authentication \(Oracle Solaris\)”](#) for details.
- Is the expected smart card bus protocol configured? See [Section 8.6.5, “How to Change the Smart Card Bus Protocol \(Oracle Solaris\)”](#) for details.
- Are the appropriate smart card configuration files available? See [Section 8.6.3, “How to Add a Smart Card Configuration File”](#) for details.
- Is the expected smart card probe order configured? See [Section 8.6.4, “How to Change the Smart Card Probe Order”](#) for details.
- Check for any core dumps in the `/tmp/SUNWut/pcscd` directory.

8.7.1. Smart Card Transaction Problems

Here are some guidelines and workarounds when using authenticated smart cards.

- When a smart card transaction is occurring on a Sun Ray Client (smart card LED is flashing), avoid hotdesking or resetting the Sun Ray Client. If a problem occurs, you may have to log out of the session.
- If you reset a Sun Ray Client that has smart card-related applications currently running, the applications may freeze for up to two minutes or the smart cards may be inaccessible for up to two minutes. The applications should recover without user intervention.

- When hotdesking with a Windows connector session running, the PIN dialog may fail to display. You will get a password prompt. If this happens, log out of the Windows session, restart the Windows connector session, and log into Windows again.
- When rapidly hotdesking a Sun Ray Client, it may fail to recognize the smart card and OSD icon 63 will display instead of the session. Remove and reinsert the card to resolve the issue.

8.8. CCID IFD Handler for External USB Smart Card Readers (Oracle Solaris)

Sun Ray Software supports the CCID IFD handler V1.3.10 on Sun Ray servers running Oracle Solaris, which provides access to external CCID-compliant USB smart card readers connected to Sun Ray Clients and client computers running Oracle Virtual Desktop Client. CCID IFD handler V1.3.10 is a Sun Ray implementation of the Interface Device Handler (IFD) for the PC/SC-lite API. When used in conjunction with the smart card services provided by the Sun Ray Software, this CCID IFD handler enables PC/SC-compliant applications and middleware to use external CCID-compliant USB smart card readers on desktop clients.

See [Section 8.6.2, “How to Configure External CCID-Compliant USB Smart Card Readers for Authentication \(Oracle Solaris\)”](#) for details on all the steps required.

8.8.1. How to Install CCID IFD Handler

Follow these instructions to install the CCID IFD handler.



Note

To install the CCID IFD handler in an Oracle Solaris Trusted Extensions environment, perform the installation as root from ADMIN_LOW (global zone).

1. Download and unpack the CCID IFD handler.

The CCID IFD Handler is not provided with the Sun Ray Software 5.4 release. However, you can download the *PC/SC-lite 1.3* component from the [5.1.1 Media Pack](#), which includes the CCID IFD Handler v1.3.10 distribution. Only the CCID IFD handler needs to be installed. PC/SC-lite is already installed with Sun Ray Software 5.4.

2. Become superuser on the Sun Ray server.
3. Install the CCID IFD handler:

```
# svcadm disable pcscd
# /usr/sbin/pkgadd -d . SUNWusb-scrdr
# svcadm enable pcscd
```

8.8.2. How to Uninstall CCID IFD Handler

Follow these instructions to remove the CCID IFD handler.



Note

To uninstall the CCID IFD handler from an Oracle Solaris Trusted Extensions environment, perform the uninstallation as root from ADMIN_LOW (global zone).

1. Become superuser on the Sun Ray server.

2. Uninstall the CCID IFD handler:

```
# svcadm disable pcscd
# /usr/sbin/pkgrm SUNWusb-scrdr
# svcadm enable pcscd
```

8.8.3. Known Issues

Here are some known issues when using external USB smart card readers.

8.8.3.1. PC/SC-lite USB Enumeration Delays

Currently, there is a delay of a few seconds before external USB readers become visible to PC/SC-lite client applications. This delay occurs whenever a PC/SC-lite instance is started for a user session as well as any other time the USB bus needs to be re-enumerated. Specifically, an enumeration delay where external USB readers are not immediately visible to an application occur under the following circumstances:

- The first time a PC/SC-lite instance is started. That is, when an application attempts to access PC/SC-lite from within a given session for the first time.
- Whenever a PC/SC-lite instance is automatically restarted after the PC/SC-lite self-terminates due to an idle period of inactivity. This is similar to the first case.
- Whenever a Session Mobility event occurs, it causes a delay in reader visibility while external USB readers on the target Sun Ray Client are re-enumerated. Session Mobility is not currently supported by the CCID IFD handler for external USB readers on Sun Rays Clients.
- Resetting or power-cycling the Sun Ray Client in a Sun Ray session.

8.8.3.2. Enumeration Delay Causes Problems for Some Applications

Certain applications, such as the Windows Smart Card login over the Windows connector, are not designed to accommodate enumeration delays associated with the USB hotplug model. Such applications do not see readers that appear after they have initially scanned the PC/SC-lite reader list. In other words, readers that appear late may be missed by an application due to any of the scenarios described above.

Sometimes applications will use the first reader they find. On Sun Ray Clients, this is invariably the internal reader, unless that reader has been disabled with the following command:

```
# utdevadm -d -s internal_smartcard_reader
```

An additional solution is to ensure that the USB reader list is visible to the application before the application scans the reader list. One way to address this is by preventing PC/SC-lite instances from timing out after a pre-specified idle period. You can disable the instance timeout by editing the `/etc/smartcard/pcscd-SunRay.conf` file and changing the `INSTANCE_TIMEOUT` parameter to -1. The shipping default value is 600 seconds (10 minutes).

When you disable inactivity timeouts by changing `INSTANCE_TIMEOUT`, PC/SC-lite instances stay around until the user's session is terminated, which can mean that many PC/SC-lite processes may be in the process table, using system resources.

We currently have no data on how much of an impact that might cause as the number of user sessions on a system grows (i.e., we have insufficient data on how that scales). In many cases, it may not be a problem at all, except that the process table will be more cluttered with inactive processes than otherwise.

Chapter 9. Hotdesking

Table of Contents

9.1. Hotdesking Overview	101
9.2. Hotdesking Without Smart Cards	101
9.2.1. NSCM and Failover Groups	102
9.2.2. How to Enable NSCM Sessions	102
9.2.3. How to Log in to an NSCM Session	103
9.3. Regional Hotdesking	104
9.3.1. Regional Hotdesking Process	105
9.3.2. Regional Hotdesking Site Requirements	105
9.3.3. Providing Site Integration Logic	105
9.3.4. How to Configure a Site-specific Mapping Library	105
9.3.5. How to Use Token Readers with Regional Hotdesking	106
9.3.6. How to Configure the Sample Data Store	106
9.4. Remote Hotdesk Authentication (RHA)	107
9.4.1. How to Disable Remote Hotdesk Authentication	108
9.4.2. How to Re-enable Remote Hotdesk Authentication	108

This chapter describes how to configure regional hotdesking and hotdesking without smart cards.

9.1. Hotdesking Overview

A Sun Ray session is a group of services controlled by a Session Manager and associated with a user through an authentication token. The sessions reside on a server rather than on the desktop. Because Sun Ray Clients are stateless, a session can be directed or redirected to any Sun Ray Client on the appropriate network or subnetwork when a user logs in or inserts a smart card.

Hotdesking, or session mobility, is the ability for a user to remove a smart card, insert it into any other client within a failover group, and have the user's session "follow" the user, thus allowing the user to have instantaneous access to the user's windowing environment and current applications from multiple clients. Every Sun Ray Client is equipped with a smart card reader, and hotdesking is configured and enabled by default.

Sun Ray Software also provides Non-Smart Card Mobility (NSCM), or hotdesking without smart cards.

9.2. Hotdesking Without Smart Cards

Configuring Sun Ray Software with non-smart card mobility (NSCM) sessions provides the benefits of hotdesking without the use of smart cards. This section explains NSCM sessions, how to configure them, and how to enable users to access their Sun Ray sessions across multiple failover groups.

NSCM can use regional hotdesking and it automatically provides similar protection as remote hotdesk authentication (RHA).

In an NSCM session, the user can:

- Type a user name and password instead of inserting a smart card.
- Type the `utdetach` command instead of removing a smart card.

If a user does not want to use the NSCM session, inserting a smart card causes the session to be disconnected and replaced by a smart card session.

9.2.1. NSCM and Failover Groups

The user login experience for NSCM sessions may be different than expected when systems are configured as part of a failover group.

The following situations might produce unfamiliar behavior:

- **Load Balancing Between Servers** - If server A is heavily loaded when a user logs into it with the NSCM GUI, the server redirects the user to server B.
- **Switching Between Servers** - A user with a session on server A who wants to switch to a session on server B invokes the `utselect` GUI to access the other session. In doing so, the user is required to log in with the NSCM GUI. Users familiar with the ease of the `utselect` GUI might be displeased that another login is necessary.
- **Escape Token Sessions** - The user bypasses the NSCM GUI by clicking the Exit button and logs into server A using the display manager (`dtlogin` or `gdm`). The user now has a standard escape token session and invokes the `utselect` GUI to switch to server B, causing the NSCM GUI to be presented again. The user must click Exit again to get to the escape token session on server B. Users accustomed to switching rapidly might find this behavior annoying.

9.2.2. How to Enable NSCM Sessions

This procedure describes how to enable NSCM sessions by using the Admin GUI or the `utpolicy` command.

Admin GUI Steps

1. Use the `utwall` command to inform your users that all active and detached sessions will be lost.

For example:

```
# /opt/SUNWut/sbin/utwall -d -t 'System policy will change in 10 minutes.  
All active and detached sessions will be lost.  
Please save all data and terminate your session now.' ALL
```

The following message is displayed in a pop-up window for all users:

```
System policy will change in 10 minutes.  
All active and detached sessions will be lost.  
Please save all data and terminate your session now.
```

2. Log in to the Admin GUI.
3. Go to the System Policy tab.
4. In the Non-Card Users panel, select the **Enabled** option next to **Mobile Sessions**.
5. Go to the Servers tab.
6. Click **Cold Restart** to restart Sun Ray services and terminate all users' sessions.

Command Line Steps

1. Use the `utwall` command to inform your users that all active and detached sessions will be lost.

For example:

```
# /opt/SUNWut/sbin/utwall -d -t 'System policy will change in 10 minutes.  
All active and detached sessions will be lost.  
Please save all data and terminate your session now.' ALL
```

The following message is displayed in a pop-up window for all users:

```
System policy will change in 10 minutes.  
All active and detached sessions will be lost.  
Please save all data and terminate your session now.
```

2. As superuser, type the `utpolicy` command with the `-M` argument for your authentication policy.

For example:

```
# /opt/SUNWut/sbin/utpolicy -a -M -s both -r both
```

This example configures the Authentication Manager to allow self-registration of users both with or without smart cards, and NSCM sessions are enabled.

3. Initialize Sun Ray services by restarting the Authentication Manager on the server, including each secondary Sun Ray server if in a failover group.

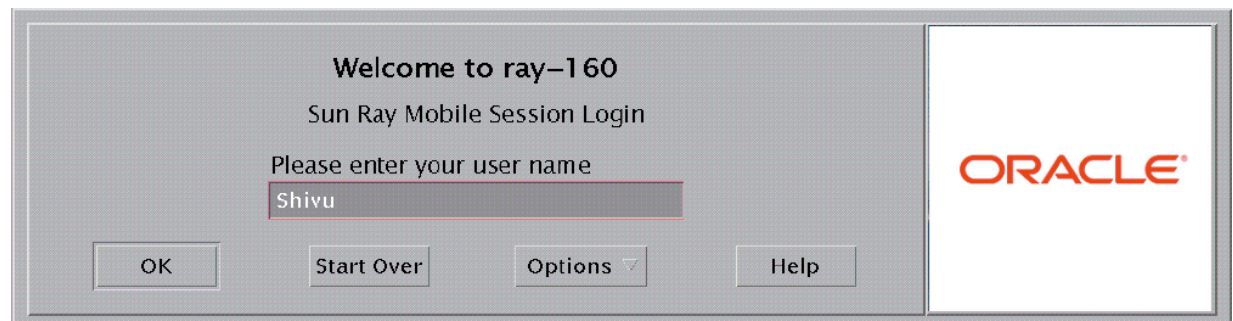
```
# /opt/SUNWut/sbin/utstart -c
```

This command clears all active and detached sessions.

9.2.3. How to Log in to an NSCM Session

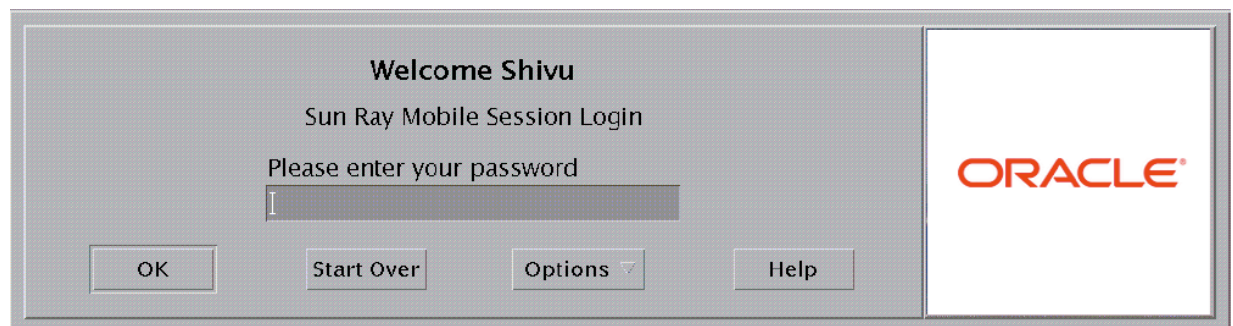
1. Type your user name into the user entry field.

Figure 9.1. NSCM Login Dialog Box User Field



2. Type your password into the password field.

Figure 9.2. NSCM Login Dialog Box Password Field



An Options menu is available for Oracle Solaris. Right clicking the Options menu displays a panel with the following options:

- **QuickLogin** - Applicable only to a new session. Selecting Off enables the user to log in with the same options available through `dtlogin`. Selecting On enables the user to bypass the option selection phase. QuickLogin is On by default. Although this option is provided with Oracle Solaris 11, it does not work with the `gdm` display manager.
- **Exit** - Selecting Exit temporarily disables the NSCM session. An escape token session is started, and the dialog box is replaced by the `dtlogin` or `gdm` screen. A user without a valid account in this server group can exit to `dtlogin` or `gdm` and attempt a remote X (XDMCP) login to some other server where that user has a valid account.

**Note**

When using Oracle Linux, the Oracle Linux login screen may briefly display before the desktop is presented. No action is necessary.

If no NSCM session exists for this user, the Authentication Manager creates an NSCM session token with the format: `mobile.IEEE802-MACID`.

9.2.3.1. Session Redirection

The user might be redirected to another server for the following reasons:

- If the Sun Ray server is part of a failover group, the load-balancing algorithm might redirect the user to another Sun Ray server.
- If the user has an NSCM session on a different Sun Ray server in a failover group, the user will be redirected to the server with the most current NSCM session.

The Sun Ray Mobile Session Login dialog box is redisplayed with the host name of the new Sun Ray server. The user must retype the password.

9.3. Regional Hotdesking

Regional hotdesking, previously referred to as Automatic Multi-Group Hotdesking (AMGH), is required when users need to move from one location to another, such as between corporate headquarters and various branch offices, and they want access to their existing session across multiple failover groups.

Multiple failover groups are useful for various reasons, such as:

- **Availability** - It is sometimes advantageous to have multiple, geographically-separate locations, each with a failover group, so that if an outage occurs at one location, another location can continue to function.
- **Organizational Policies** - Some sites have different administrative policies at different locations. It can be advantageous to keep separate failover groups at these locations.

For further technical detail, please refer to the `utamghadm(8)`, `ut_amgh_get_server_list(3)`, and `ut_amgh_script_interface(3)` man pages.

**Note**

Regional hotdesking is not enabled for multihead groups.

9.3.1. Regional Hotdesking Process

Once regional hotdesking is configured, user login information and sessions are handled as follows:

1. When a smart card is inserted or removed from the system or a user logs in via the greeter GUI, parameters such as the user name (if known at the time), smart card token, and terminal identifier are passed to a piece of site integration logic.
2. The site-integration software uses these parameters to determine to which Sun Ray servers it should direct the Sun Ray Client.
3. If the smart card token is associated with a local session, then that session gets preference, and regional hotdesking is not invoked.
4. Otherwise, the regional hotdesking software redirects the Sun Ray Client to connect to the appropriate Sun Ray server.

Thus, if the user has an existing session, the client connects to that session; if not, the regional hotdesking software creates a new session for that user.

9.3.2. Regional Hotdesking Site Requirements

To use regional hotdesking, a site must provide some site integration logic that can utilize enterprise data to determine which users or Sun Ray Clients should connect to which failover groups. This is ordinarily provided through the use of a dynamic C library or a shell script that implements a particular interface used by regional hotdesking software. Sun Ray Software provides some reference code that you use as an example or adapt as required. An administrator must configure the regional hotdesking software to utilize a specified library or shell script, then implement the PAM stack of the login applications, as described below.



Note

To ensure continuous operation, be sure to include enough servers in the target group to provide availability for session location and placement in the event that a particular server becomes unavailable. Two servers should be minimally sufficient for most sites; three servers provide a conservative margin of error.

9.3.3. Providing Site Integration Logic

To determine where given Sun Ray Clients or users should be connected when creating or accessing sessions, you must utilize enterprise data. Sun Ray Software includes the following software for this purpose:

- Man pages, such as `ut_amgh_get_server_list(3)`, which describe the appropriate C API for a shared library implementation.
- A shell-script API, `ut_amgh_script_interface(3)`, which can be used as an alternative.
- Reference C code and script code, located at `/opt/SUNWutref/amgh`. This code can serve as example or be directly adapted for use.
- A functional Makefile.

9.3.4. How to Configure a Site-specific Mapping Library

For each site, you must determine what mapping library to use. It may be a site-specific implementation, or one of the sample implementations provided with the Sun Ray Software.

**Note**

If you are using Oracle Linux, library mapping for the 32-bit platform should be `/opt/SUNWutref/amgh/lib`, as shown below, and library mapping for the 64-bit platform should be `/opt/SUNWutref/amgh/lib64`.

After configuring the library, you must perform a cold restart of the Sun Ray services using either the `utstart` CLI or the Admin GUI.

- How to Configure the Token-based Mapping Implementation Provided as a Sample

```
# /opt/SUNWut/sbin/utamghadm -l /opt/SUNWutref/amgh/lib/libutamghref_token.so
```

- How to Configure the User Name-based Mapping Implementation Provided as a Sample

```
# /opt/SUNWut/sbin/utamghadm -l /opt/SUNWutref/amgh/lib/libutamghref_username.so
```

- How to Configure a Script-based Back-end Mapping

```
# /opt/SUNWut/sbin/utamghadm -s /opt/SUNWutref/amgh/utamghref_script
```

9.3.5. How to Use Token Readers with Regional Hotdesking

To utilize token readers with regional hotdesking based on Sun Ray pseudo-tokens, use the Site-specific Mapping Library to produce the desired behavior for them.

Configured token readers should have the following value formats:

Key	Value
<code>insert_token</code>	<code>pseudo.MAC_address</code>
<code>token</code>	<code>TerminalId.MAC_address</code>

If a registered policy is in place, use the `insert_token` key instead of the token key, which is not globally unique.

**Note**

The RHA security feature does not affect token readers. It is assumed that token readers are deployed in physically secure environments.

9.3.6. How to Configure the Sample Data Store

Each site must configure a data store to contain site-specific mapping information for regional hotdesking. This data store is used by the site mapping library to determine whether regional hotdesking should be initiated for the parameters presented. The data store can be a simple flat file. The sample implementations included with the Sun Ray Software require a simple flat file configuration.

To create the back-end database file under `/opt/SUNWutref/amgh/back_end_db` on the Sun Ray server, do the following:

- For a token-based mapping, use entries of the form:

```
token=XXXXXXX [username=XXXXX] host=XXXXX
```

- Comments (lines beginning with #) are ignored.

- `username` is optional. If the same token is associated with more than one non-null `username`, an error is returned.
- For a user name-based mapping, use entries of the form:

```
username=XXXXX host=XXXXX
```

- Comments (lines beginning with #) are ignored.
- Key/value pairs other than those mentioned above are ignored.
- The order of key/value pairs is not significant.

- For a combined mapping, use entries of the form:

```
Any combination of TOKEN BASED and USERNAME BASED lines.
```

- Comments (lines beginning with #) are ignored.
- A token match is attempted first.
- If no token match is made (or if no `username` is included in the matches) the user is prompted for a `username`.
- A lookup is made for this `username`. If there is no match, a local session is created; otherwise, the Sun Ray Client is forwarded to the first host reported as available.

A sample line for this file would look like the following:

```
token=MicroPayflex.5001436700130100 username=user1 host=ray-207
```

9.4. Remote Hotdesk Authentication (RHA)

By default, when a user hotdesks, the desktop's screen lock is activated and the user is forced to authenticate again. However, screen locks are inherently insecure in a number of ways. Remote Hotdesk Authentication (RHA) is designed to provide a more secure hotdesk environment instead of the authentication performed by a desktop screen lock in the user's existing session. The "Remote" in RHA refers to the fact that the hotdesk authentication step takes place outside the user's existing session and applications cannot interfere with the authentication. From a user's perspective, there is minimal change if Remote Hotdesk Authentication is enabled.

When RHA is enabled and a reconnection is attempted, the Sun Ray Software creates a temporary new session for the client and uses that session to present an authentication dialog to the user. (This RHA dialog looks very similar to the NSCM authentication dialog.) After the user successfully authenticates through the dialog, the temporary session is dismissed and the user's existing session is connected to the client.

For environments where the in-session screen lock provides acceptable security or where no hotdesk authentication is desired, you can configure Sun Ray Software to disable the RHA security feature.

RHA is enabled for smart cards by default and NSCM automatically provides similar protection as RHA. Authentication does not apply to anonymous Kiosk Mode.



Note

The RHA security feature does not affect token readers. It is assumed that token readers are deployed in physically secure environments.

9.4.1. How to Disable Remote Hotdesk Authentication



Note

Disabling the RHA feature may present a security risk under some circumstances.

1. To disable RHA configuration for a group, type the following command:

For example, if your policy allows smart cards and non-smart card logins and failover groups, use the following command and options to disable RHA:

```
# utpolicy -a -z both -g -D
```

2. Perform a cold restart of the Sun Ray services:

```
# utstart -c
```

9.4.2. How to Re-enable Remote Hotdesk Authentication

1. Restate your policy using `utpolicy` without the `-D` option.

For example, to reinstate a policy that allows smart cards and non-smart card logins and failover groups with RHA, use the following command and options:

```
# utpolicy -a -z both -g
```

2. Perform a cold restart of the Sun Ray services:

```
# utstart -c
```

Chapter 10. Kiosk Mode

Table of Contents

10.1. Kiosk Overview	109
10.2. Kiosk Mode Security and Failover Considerations	110
10.3. Kiosk User Accounts	110
10.3.1. Characteristics	110
10.3.2. Restrictions and Safe Guards	111
10.3.3. Administering the Kiosk User Pool	111
10.4. Session Type Components	111
10.4.1. Session Descriptor	111
10.4.2. Session Script	112
10.5. How to Configure Kiosk Mode and User Accounts	112
10.6. How to Add Kiosk User Accounts	112
10.7. How to Configure a Kiosk Mode Session Type	113
10.8. How to Enable and Disable Kiosk Mode	116
10.8.1. Unconfiguring Kiosk Mode Disables Kiosk Policy	117
10.9. How to Override the Default Kiosk Mode Policy	117
10.10. Configuring the Windows Connector Kiosk Session Type	119
10.10.1. How to Configure a Kiosk Mode Session Type for the Windows Connector	120
10.11. Configuring the VMware View Connector Kiosk Session Type	122

This chapter describes the kiosk mode functionality provided by the Sun Ray Software.

10.1. Kiosk Overview

Sun Ray Software offers two modes of operation:

- **Regular mode** - Delivers a desktop based on the platform that is installed on the Sun Ray server. For instance, if the Sun Ray Software is installed on Oracle Linux, users get an Oracle Linux desktop. Users authenticate on this platform and traditionally execute applications located on the platform.
- **Kiosk mode** - Delivers an almost unlimited variety of desktops or applications to users, even though the actual desktop or application may be running elsewhere. Kiosk mode bypasses the normal authentication methods of the Sun Ray server platform and runs anything that the administrator defines. By creating customized session types, a kiosk session can be anything from a full screen instance of a Firefox web browser to a full screen Windows 7 desktop running in a virtual machine.

By default, Sun Ray Software provides the following pre-defined session types that you can configure for users.

- **Sun Ray Connector for Windows OS** - Provides a windows desktop through the *Windows connector*.
- **VMware View Manager Session** - Provides a Windows virtual machine through the *VMware View connector*.
- **Sun Java Desktop System 3** - Provides a locked-down Oracle Solaris 10 desktop environment that can run other applications (Oracle Solaris 10 Sun Ray server only).
- **Common Desktop Environment (Obsolete)** - Provides a locked-down Oracle Solaris 10 desktop environment that can run other applications (Oracle Solaris 10 Sun Ray server only).

Kiosk mode can also provide unauthenticated access for settings such as public kiosks, where users cannot be expected to provide authentication credentials.

10.2. Kiosk Mode Security and Failover Considerations

Because kiosk mode bypasses the system login mechanism, you must consider the security of the applications added to the user environment. Many custom applications provide built-in security, but applications that do not are not suitable for kiosk mode.

For example, adding an application such as `xterm` provides users with access to a command-line interface from a kiosk mode session. This access is not desirable in a public environment and is not advised. However, using a custom application for a call center is perfectly acceptable.

In a failover environment, the kiosk mode administrative settings are copied from the primary server to the secondary servers. Be sure that all application descriptors and executable paths added to the kiosk mode sessions are copied across the servers in the failover group. For example, if a Mozilla application is added to the sessions with the executable path `/usr/sfw/bin/mozilla`, make sure that the path to the binary is available to all servers in the failover group.

One way to ensure that sessions and applications are available on all servers in a failover group is to put them into a shared network directory, which is available on all hosts in the failover group. You can do this through a highly available file share, such as the [Oracle Solaris Cluster product](#).

10.3. Kiosk User Accounts

All computer applications must run under some type of user account and kiosk sessions are no different. To enable real users to access applications without requiring the need to authenticate to the underlying operating system of the Sun Ray Software, kiosk mode manages a pool of local user accounts. If the kiosk service determines that an administrator has configured the system policy or the current token ID to run a kiosk session, unauthenticated access to the system is granted.

While kiosk user accounts do not correspond to a real user, their role in kiosk mode allows a real user to use the applications defined by the administrator in a unauthenticated manner. Without a kiosk user account, a kiosk session cannot run.

See [Section 10.5, “How to Configure Kiosk Mode and User Accounts”](#) for details on setting up kiosk mode user accounts.

10.3.1. Characteristics

Kiosk user accounts have the following characteristics:

- A default naming scheme of `utkux`, where `x` is a range from 0 to `N-1` and `N` is the specified number of kiosk user accounts to create.
- A different naming prefix can be chosen if the default of `utku` has risk of a collision. If you need to change the existing kiosk mode user accounts due to a collision problem, you can use the `kioskuseradm` command.
- UID by default starts at 150000 (starting UID can be specified).
- UID range must be contiguous.
- Home directories are located in `/var/opt/SUNWkio/home/$USER`.
- Local accounts only (`/etc/passwd`). Centralized NIS or LDAP kiosk user accounts are not supported.

10.3.2. Restrictions and Safe Guards

To limit the impact a kiosk user can have on the system and prevent unauthenticated access from becoming uncontrolled access, the following restrictions and safe guards are placed on kiosk user accounts:

- Kiosk user accounts are locked for normal logins (GDM, SSH, Telnet, etc).
- Kiosk user accounts belong to a local Unix group (`utkiosk`) that has minimal rights on the system.
- No two sessions use the same kiosk user account at the same time on the same server.
- The home directory associated with a kiosk user account is completely cleared after a session ends.
- The home directory associated with a kiosk user account is created and then populated from the `prototypes` directory when a session is started.
- Residual processes owned by the kiosk user account are killed when a kiosk session ends and before a new session is started.
- All files in the `/tmp` and `/var/tmp` directories owned by the kiosk user account are deleted when a kiosk session ends and before a new session is started.

10.3.3. Administering the Kiosk User Pool

If you need to change the number of kiosk user accounts after the initial Sun Ray Software installation and configuration, you can manage the user pool after the initial configuration using the `kioskuseradm` command located in the `/opt/SUNWkio/bin` directory. With this command, you can view the pool settings, see how many kiosk user accounts are in use, and modify the pool settings such as adding or decreasing the number of kiosk user accounts. You can increase or decrease the pool of users while kiosk sessions are active, but changing other pools settings such as group membership or UID range requires that no kiosk sessions are active.



Note

The kiosk user accounts in the kiosk user pool must have contiguous user IDs. If any user accounts were added after the configuration of the initial pool of kiosk user accounts, you cannot use the `kioskuseradm extend` command. The `extend` option relies on kiosk user accounts with contiguous user IDs.

To work around this issue, you must delete all the kiosk user accounts and recreate them by using the `kioskuseradm modify` command. This process requires you to stop the Sun Ray services on the Sun Ray server. If you have a failover group, performing these steps on each Sun Ray server separately will avoid user downtime.

10.4. Session Type Components

A session type is a collection of scripts and files used to deliver a kiosk session to a user, such as the pre-defined Sun Ray Connector for Windows OS session type. Each session type has a session descriptor and a session script, which are located in the `/etc/opt/SUNWkio/sessions` directory.

10.4.1. Session Descriptor

The session descriptor defines a number of attributes useful for the administration and launching of the session. [Table 10.1, “Kiosk Session Descriptors”](#) provides the list of key session descriptors.

Table 10.1. Kiosk Session Descriptors

Kiosk Session Descriptors	Descriptor Description
<code>KIOSK_SESSION_EXEC</code>	Identifies the location of the session script.
<code>KIOSK_SESSION_LABEL</code>	Identify a label and description respectively to be used by the Admin GUI.
<code>KIOSK_SESSION_DESCRIPTION</code>	
<code>KIOSK_SESSION_ARGS</code>	Identifies default session script arguments.

10.4.2. Session Script

The session script is used to launch the session type, and it is usually a simple wrapper of an executable.

Both `uttsc` and specific session type arguments are accepted by session scripts, and you can specify them through the Admin GUI when configuring the session type. Any `uttsc` arguments are simply passed directly to the Windows connector and not processed by the session script. Other arguments specific to the session type are processed by the session script. Each kiosk session type supports a different set of arguments.

10.5. How to Configure Kiosk Mode and User Accounts

As part of the initial configuration during the Sun Ray Software installation, you are given the opportunity to initially configure kiosk mode, which consists of configuring the kiosk user accounts. See [Section 10.3, “Kiosk User Accounts”](#) for more information.

If you don't initially configure kiosk mode, you can always use the `utconfig -k` command to configure it later. You can also perform additional kiosk mode account management tasks by using the `kioskuseradm` command.

10.6. How to Add Kiosk User Accounts

This procedure describes how to add more kiosk user accounts to the user account pool. You can increase the number of kiosk user accounts even while there are existing kiosk sessions.

See [Section 10.3, “Kiosk User Accounts”](#) for more details.



Note

The kiosk user accounts in the kiosk user pool must have contiguous user IDs. If any user accounts were added after the configuration of the initial pool of kiosk user accounts, you cannot use the `kioskuseradm extend` command. The `extend` option relies on kiosk user accounts with contiguous user IDs.

To work around this issue, you must delete all the kiosk user accounts and recreate them by using the `kioskuseradm delete` and `kioskuseradm create` commands, respectively. This process requires you to stop the Sun Ray services on the Sun Ray server. If you have a failover group, performing these steps on each Sun Ray server separately will avoid user downtime.

1. Become superuser on the Sun Ray server.
2. Increase the number of kiosk user accounts.

```
# /opt/SUNWkio/bin/kioskuseradm extend -c number_of_new_users
```

10.7. How to Configure a Kiosk Mode Session Type

This procedure describes how to configure a kiosk mode session type, which determines what type of session is launched in kiosk mode.

You can use the Admin GUI to configure a kiosk mode session type. On the Kiosk Mode tab, accessed from the Advanced tab, you can choose a predefined session type. You can also specify other general properties that control kiosk mode behavior, such as Timeout, Maximum CPU Usage, and Maximum VM Size.

Some session types allow additional kiosk applications to be launched. Not all session types support this ability. For example, a kiosk full-screen web browser session does not need to have this capability. The applications table on the Kiosk Mode tab page is displayed or hidden depending on what session type is selected.

You can add a new kiosk application by clicking the New button in the applications table and specify it using either a predefined application descriptor file or by specifying the path to an executable or an application descriptor on the server. All predefined application descriptors are located in the `/etc/opt/SUNWkio/applications` directory. For a detailed explanation of kiosk mode functionality, see the `kiosk` man page.

To configure the pre-defined session types, the following procedures are provided:

- [Section 10.10.1, “How to Configure a Kiosk Mode Session Type for the Windows Connector”](#)
- [Section 18.4.1, “How to Configure a Kiosk Mode Session Type for the VMware View Connector”](#)

Admin GUI Steps



Note

Kiosk session and application configuration data created with the Admin GUI is stored as the default kiosk session type under the name `session`. To store non-default kiosk session types, use the `utkiosk` command on the command line.

1. Click the Advanced tab.
2. Click the Kiosk Mode tab from the Advanced tab, as shown in [Figure 10.1, “Edit Kiosk Mode Screen”](#).

Figure 10.1. Edit Kiosk Mode Screen

3. Click **Edit**.
4. Select your preferred **Session** (Session Type) from the drop-down list.
5. Provide appropriate values for the remaining settings, which are described in [Table 10.2, “ulimit Settings”](#). For more information, see the [ulimit](#) man page.

**Note**

Choosing unsuitable values for `ulimit` settings could cause kiosk sessions to start incorrectly or to crash due to lack of resources.

Table 10.2. `ulimit` Settings

Value	Description
Timeout	Indicates the number of seconds after which a disconnected session will be terminated. If you provide no value for this setting, termination of disconnected sessions will be disabled.
Maximum CPU Time	Indicates the maximum number of CPU seconds per process for kiosk sessions. By default, the system default is applied to all kiosk sessions.

Value	Description
Maximum VM Size	Indicates the maximum Virtual Memory size per process for kiosk sessions. By default, the system default is applied to all kiosk sessions.
Maximum Number of Files	Indicates the maximum number of open files per process for kiosk sessions. By default, the system default is applied to all kiosk sessions.
Maximum File Size	Indicates the maximum file size per process for kiosk sessions. By default, the system default is applied to all kiosk sessions.
Locale	Indicates the locale to be used by the kiosk session. By default, the system default is applied to all kiosk sessions.
Arguments	Indicates a list of arguments that should be passed to kiosk sessions as they start. This setting is specific to the kiosk session. For more information about supported arguments, consult the session-specific documentation for your selected session.

6. Click **OK**.

Changes to the kiosk mode settings are applied automatically to kiosk sessions started after you configure the session type. You do not have to restart Sun Ray services for changes to take effect.

Command Line Steps

1. Create a session configuration file.

- a. To start with an existing configuration, export the settings to a file. For example:

```
utkiosk -e session -s > mysession.conf
```

- b. Edit the `mysession.conf` file.

See the `session.conf` man page for a description of available settings. The following example uses the Sun Ray Windows Connector kiosk session:

```
KIOSK_SESSION=uttsc
KIOSK_SESSION_LIMIT_VMSIZE=20000
KIOSK_SESSION_ARGS=-h -- -r sound:low -E theming winserver.example.org
```

2. If applicable, create an application list file.

If you are using a kiosk session that can serve as a container for multiple applications, you should create an application list file.

- a. To start with existing settings, export the application list to a file:

```
utkiosk -e session -a > myapps.list
```

- b. Edit the `myapps.list` file.

See the `kiosk` man page for a description of application list files.

3. Import your settings into the Sun Ray data store.

- To import your session settings without an application list as the default session configuration:

```
utkiosk -i session -f mysession.conf
```

- To import your session settings and application list as the default session configuration:

```
utkiosk -i session -f mysession.conf -A myapps.list
```

- To import your session settings as non-default session configuration:

```
utkiosk -i MySpecialSession -f mysession.conf
```

10.8. How to Enable and Disable Kiosk Mode

Kiosk mode can be enabled as the default session type for smart card users, non-smart card users, or both. When kiosk mode is enabled for a class of tokens, this choice can be overridden for individual tokens. For example, when kiosk mode is enabled for card users, regular non-kiosk session access can be configured for individual cards. Alternatively, a kiosk session other than the default kiosk session can be configured for individual tokens. Enabling and disabling kiosk mode for individual tokens is described in [Section 10.9, “How to Override the Default Kiosk Mode Policy”](#).

Before enabling kiosk mode, you must configure the kiosk mode user accounts.

Admin GUI Steps

Kiosk mode functionality can be enabled and disabled from the System Policy section of the Advanced tab, which provides options to enable kiosk mode for smart card users, non-smart card users, or both.

Command-Line Steps

1. Become superuser on the Sun Ray server.
2. Enable a kiosk mode through the `utpolicy -k` command.

The following options determine whether access to the Sun Ray server is granted to certain tokens:

```
-z both/pseudo/card
```

or

```
-r both/pseudo/card [-s both/pseudo/card]
```

The `-k both/pseudo/card` option determines whether some or all of the granted sessions are kiosk sessions.

How to Enable Kiosk Mode for All Users (Smart Card and Non-Smart Card)

```
# /opt/SUNWut/sbin/utpolicy -a -M -s both -r both -k both
```

All users are directed to kiosk sessions.

How to Allow Only Smart Card Sessions in Kiosk Mode

```
# /opt/SUNWut/sbin/utpolicy -z card -k card
```

All sessions are in kiosk mode and available only to smart card users unless you specify overrides.

How to Enable Kiosk Mode for Smart Card Users Only

```
# /opt/SUNWut/sbin/utpolicy -a -M -s both -r both -k card
```

Only smart card users are directed to kiosk sessions.

How to Enable Kiosk Mode for Non-Smart Card Users Only

```
# /opt/SUNWut/sbin/utpolicy -a -s both -r both -k pseudo
```

Only non-smart card users are directed to kiosk sessions.

How to Enable Regular Sessions for Smart Card Users and Kiosk Sessions for Non-Smart Card Users

```
# /opt/SUNWut/sbin/utpolicy -z both -k pseudo
```

Smart card sessions are non-kiosk (ordinary login) sessions. Non-smart card sessions are kiosk sessions.

How to Enable Regular Sessions for Registered Smart Cards and Kiosk Sessions for Non-Smart Card Users

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

Non-kiosk smart card sessions are allowed only for registered tokens. Non-smart card sessions are kiosk sessions.

How to Enable Kiosk Sessions for Registered Smart Cards and Regular Sessions on Registered Clients

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

Smart card sessions are kiosk sessions, non-smart card sessions are non-kiosk (ordinary login) sessions. Users can self-register smart card tokens and clients.

How to Allow Only Card Sessions in Kiosk Mode

```
# /opt/SUNWut/sbin/utpolicy -z card -k card
```

All sessions are in kiosk mode and available only to smart card users unless you specify overrides.

10.8.1. Unconfiguring Kiosk Mode Disables Kiosk Policy

If Kiosk mode is enabled for smart card and/or for non-card sessions, then disabling Kiosk mode (using `utconfig -u -k`) also disables the Kiosk policy.

This behavior may be surprising in a failover group, where the Kiosk policy is disabled for the entire group when Kiosk Mode is unconfigured on any server in the group.

Before unconfiguring Kiosk Mode on any host in a failover group, disable the Kiosk policy, and perform a cold restart of the server group.

To perform maintenance tasks on Kiosk user accounts without unconfiguring Kiosk Mode completely, use the `/opt/SUNWkio/bin/kioskuseradm` tool instead of `utconfig`.

10.9. How to Override the Default Kiosk Mode Policy

You might need to assign a different authentication policy setting for a particular smart card or Sun Ray Client, or subset of smart cards or Sun Ray Clients. Only registered tokens can be assigned policy overrides.

Admin GUI Steps



Note

The Edit Token Properties page does not show whether a non-default kiosk session has been assigned to a token. If you use the Admin GUI to assign a kiosk session type to a token, the default kiosk session configuration is used for that token.

1. Click the Tokens tab.
2. Select the token of interest from the list of tokens.

This token can be a card owner's smart card token or a pseudo-token associated with a client's MAC address. However, only registered tokens can be overridden. For more information, see [Chapter 7, Sessions and Tokens](#).

3. Click **Edit**.

Figure 10.2. Edit Token Properties Screen

The screenshot shows the Oracle Sun Ray Administration interface. At the top, there's a header with 'ORACLE Sun Ray Administration' and user information: 'User: admin Server: oel55-paul'. Below the header are tabs: 'SERVERS', 'SESSIONS', 'DESKTOP UNITS', 'TOKENS' (selected), 'ADVANCED', and 'LOG FILES'. The main content area is titled 'Tokens > MicroPayflex.5001430700130100' and 'Edit Token Properties - MicroPayflex.5001430700130100'. There are 'OK' and 'Cancel' buttons. A note says '* Indicates required field'. The 'General' section has fields for '* Owner: Demo', 'Other Information: Test token', and 'Status: ☒ Enabled'. The 'Advanced' section has a 'Session Type' dropdown menu with options: 'Default', 'Kiosk', and 'Regular'. There are 'OK' and 'Cancel' buttons at the bottom right.

4. Select the desired **Session Type** from the list of available session types.

The available session types are **Default**, **Kiosk**, and **Regular**.

- Select **Default** to prevent the kiosk mode policy from being overridden for this token.

- Select **Kiosk** to use a kiosk session for this token regardless of the kiosk mode policy.
 - Select **Regular** to ensure that a kiosk session is not used for this token regardless of the kiosk mode policy.
5. Click OK.

Command-Line Steps

- Use the `utkioskoverride` command to override the policy.

```
/opt/SUNWut/sbin/utkioskoverride
```

The following examples demonstrate how to override the kiosk mode policy from the command line. For more detailed information about overriding kiosk mode policy, see the `utkioskoverride` man page.

How to Enable Kiosk Sessions Regardless of the Kiosk Mode Policy for a Registered Smart Card

To enable kiosk sessions regardless of the kiosk mode policy for the registered smart card `MicroPayFlex.12345678`:

```
# /opt/SUNWut/sbin/utkioskoverride -s kiosk -r MicroPayFlex.12345678
```

How to Disable Kiosk Session Regardless of the Kiosk Mode Policy for a Registered Smart Card

To disable kiosk sessions regardless of the kiosk mode policy for the registered smart card `MicroPayFlex.12345678`:

```
# /opt/SUNWut/sbin/utkioskoverride -s regular -r MicroPayFlex.12345678
```

How to Disable Kiosk Sessions Regardless of the Kiosk Mode Policy for a Logical Token

To disable kiosk sessions regardless of the kiosk mode policy for the logical token `user.12345678`:

```
# /opt/SUNWut/sbin/utkioskoverride -s regular -t user.12345678
```

How to Assign and Enable a Non-Default Kiosk Session

To assign and enable the non-default kiosk session `MySession2`, stored using `utkiosk`, to the logical token `user.12345678`, regardless of the kiosk mode policy:

```
# /opt/SUNWut/sbin/utkioskoverride -s kiosk -c MySession2 -t user.123456-78
```

10.10. Configuring the Windows Connector Kiosk Session Type

By using kiosk mode, you can set up a Sun Ray Client to behave just like a Windows system, which means users will not have to interact with the Oracle Solaris or Oracle Linux login screen and no longer need to specify the `uttsc` command.

Sun Ray Software provides a pre-defined Windows connector session type (called the Sun Ray Connector for Windows OS), which can be set up to provide the Windows connector kiosk session to users. The core components of the Windows connector session type are:

- Session descriptor - `/etc/opt/SUNWkio/sessions/uttsc.conf`
- Session script - `/etc/opt/SUNWkio/sessions/uttsc/uttsc`

Added applications are not supported.

The `/opt/SUNWuttsc/bin/uttsc` script is used to launch the Windows connector. This script provides a simple wrapper for the `uttsc` executable.

A two-minute timeout is imposed on Windows sessions that remain at the Windows login screen. When this timeout elapses, the associated Windows session is terminated and the Windows connector terminates subsequently. If no Windows login takes place, the client appears to reset every two minutes.

To avoid the two-minute timeout, the session script supports its own timeout mechanism, which is initiated when the script detects that the Windows connector has terminated. If the session script timeout interval has not elapsed, the session script relaunches the Windows connector. If the session script timeout has elapsed, the session script terminates, and the Kiosk session also terminates as a result. The timeout may be specified as a session script argument. It has a default value of 30 minutes.

10.10.1. How to Configure a Kiosk Mode Session Type for the Windows Connector

1. Log in to the Admin GUI.
2. Click the Advanced tab and Kiosk Mode sub-tab. Then click **Edit**.
3. Choose **Sun Ray Connector for Windows OS** from the **Session** (Session Type) menu, as shown in [Figure 10.3, “Edit Kiosk Mode Screen for Windows Connector”](#).

Figure 10.3. Edit Kiosk Mode Screen for Windows Connector

The screenshot displays the Oracle Sun Ray Administration web interface. At the top, the header includes the Oracle Sun Ray Administration logo and navigation links for Version, Log Out, and Help. Below the header, a series of tabs are visible: SERVERS, SESSIONS, DESKTOP UNITS, TOKENS, ADVANCED, and LOG FILES. The ADVANCED tab is selected, and within it, the Kiosk Mode sub-tab is active. The main content area is titled 'Edit Kiosk Mode' and includes a brief instruction: 'Specify the session type and general properties for Kiosk Mode. Click OK to store the changes.'

The configuration form contains the following fields:

- Session:** A dropdown menu with 'Sun Ray Connector for Windows OS' selected.
- Timeout:** A text input field containing '12000' with a 'seconds' label.
- Maximum CPU Time:** A text input field with a 'seconds' label.
- Maximum VM Size:** A text input field with a 'KB' label.
- Maximum number of Files:** A text input field.
- Maximum File Size:** A text input field with a '512B blocks' label.
- Locale:** A text input field.
- Arguments:** A large text input field with the default value '-t 1800 -b' displayed below it.

At the bottom right of the form, there are two buttons: 'OK' and 'Cancel'.

**Note**

Once the Windows connector session is selected, most of the fields on the main Kiosk page are not available. The Applications list is not available because the Windows connector session does not support the addition of applications.

4. Add session arguments to the Arguments field at the bottom in the format:

```
[session-type-arguments] [-- uttsc-arguments] myhost.mydomain
```

See [Table 10.3, “Kiosk Session Arguments for Windows Connector”](#) for the list of valid session arguments.

Table 10.3. Kiosk Session Arguments for Windows Connector

Argument	Description
<code>-t <i>timeout</i></code>	Sets the value of a timeout interval (in seconds) after which the session script will terminate in the event of a Windows connector termination. If Windows connector terminates before the timeout has elapsed it will be restarted by the session script. The default value for <i>timeout</i> is 1800 (30 minutes). Values less than or equal to 0 indicate that the session script should never restart the Windows connector.
<code>-h</code>	Disables the default behavior of starting <code>uttsc</code> with the <code>-m -b</code> options. Only the <code>-m</code> option is used, which means both full screen mode and the pull-down header are enabled.
<code>-- <i>uttsc-arguments</i></code> <code><i>myhost.mydomain</i></code>	Specify any valid <code>uttsc</code> arguments. For detailed information on these options, refer to the <code>uttsc</code> man page. The <code>-m</code> and <code>-b</code> <code>uttsc</code> arguments are used by default. These arguments enable full-screen mode and disable the pull-down header respectively. USB redirection is also enabled by default. The minimal required argument is the host name, so the field should contain, at minimum, <code>myhost.mydomain</code> .

The following example line specifies:

- A 10-minute timeout (specified in seconds) until the session is cycled if the user does not log in
 - Printer forwarding
 - Smart card redirection
 - Optimized Windows connector hotdesking behavior

```
-t 600 -- -r printer:officelaser -r scard:on -O myhost.mydomain.com
```

5. Configure the server to use kiosk mode for card and non-card users.
 - a. Click the System Policy sub-tab on the Advanced menu.
 - b. Enable **Kiosk Mode** for both card and non-card users.
6. Click **Save**.

All new or restarted sessions matching the policy configuration to use kiosk mode will access the new session type.

10.11. Configuring the VMware View Connector Kiosk Session Type

See [Section 18.4, “Configuring the VMware View Connector Kiosk Session Type”](#) for details.

Chapter 11. Client-Server Security

Table of Contents

11.1. Client-Server Security Overview	123
11.2. Encryption and Authentication	124
11.2.1. Security Modes	124
11.2.2. How to Force Encryption	125
11.2.3. How to Force Server Authentication	125
11.2.4. How to Disable Client Authentication	126
11.2.5. How to Force Client Authentication From All Clients	126
11.3. Managing Client Keys	127
11.3.1. Key Fingerprint	128
11.3.2. How to Deny Access to Clients With Unconfirmed Keys	128
11.3.3. How to Confirm a Specific Client Key	128
11.3.4. How to Confirm All Unconfirmed Client Keys	129
11.3.5. How to Display a Client's Fingerprint Key from a Sun Ray Client	129
11.3.6. How to Display All Client Keys	130
11.3.7. How to Display All Keys for a Specific Client	130
11.3.8. How to Delete a Specific Client Key	131
11.3.9. How to Delete All Client Keys for a Specific Client	131
11.4. Displaying Security Status	131
11.4.1. How to Display Security Status for a Sun Ray Client	131
11.4.2. How to Display Security Status for All Sessions	131
11.5. Authentication Troubleshooting	132
11.5.1. Error Messages	132

This chapter provides details on the security aspects of the interactions between the desktop clients and the Sun Ray server.

For more information about all the security aspects of Sun Ray Software, refer to the [Sun Ray Software 5.4 Security Guide](#).

11.1. Client-Server Security Overview

When configuring the security for a Sun Ray environment, you should evaluate the security requirements. You can choose one of the following security policies between the Sun Ray server and clients:

- Enable encryption for upstream traffic only (client to server)
- Enable encryption for downstream traffic only (server to client)
- Enable bidirectional encryption
- Enable server authentication
- Disable client authentication

Additionally, you must decide whether to enable hard security mode for encryption and client authentication.

You can use the `utcrypto` command or the Admin GUI to configure the encryption option, authentication option, and security mode.

11.2. Encryption and Authentication

By default, data packets between the Sun Ray server and client are sent "in the clear." This policy means that outsiders can easily "snoop" the traffic and recover vital and private user information, which malicious users might misuse. To avoid this type of attack, Sun Ray Software administrators can enable traffic encryption through the ARCFOUR encryption algorithm.

The ARCFOUR encryption algorithm, selected for its speed and relatively low CPU overhead, supports a higher level (128-bit) of security between Sun Ray services and clients.

However, encryption alone does not provide complete security. Spoofing a Sun Ray server or a Sun Ray Client and posing as either is still possible, if not necessarily easy. Here are some examples:

- A man-in-the-middle attack, in which an impostor claims to be the Sun Ray server for the clients and pretends to be the client for the server. The impostor then intercepts all messages and has access to all secure data.
- Manipulating a client to pretend to be another client in order to gain access to sessions connected to the spoofed client.

Server and client authentication provided by Sun Ray Software can resolve these types of attacks. Server authentication uses a single pre-configured, public-private key pair in the Sun Ray Software and firmware, and client authentication uses an automatically generated public-private key pair in every client.

Sun Ray Software uses the Digital Signature Algorithm (DSA) to verify that clients are communicating with a valid Sun Ray server and that the server is communicating with a legitimate client. This authentication scheme is not completely foolproof, but it mitigates trivial man-in-the-middle attacks and makes spoofing Sun Ray servers or Sun Ray Clients harder for attackers.

Enabling encryption and authentication is optional. The system or network administrator can configure it based on site requirements. By default only client authentication is enabled.

11.2.1. Security Modes

When you configure encryption and client authentication, you must decide between hard and soft security modes. Security mode can be configured separately for encryption requirements including server authentication and for client authentication requirements. Security mode settings are intended for compatibility with older firmware, which did not support the affected security feature.

- **Hard Security Mode** - Hard security mode ensures that every session is secure. If security requirements cannot be met, the session is refused.
- **Soft Security Mode** - Soft security mode ensures that connection requests are granted even for Sun Ray Clients that don't support the configured security requirements. If security requirements cannot be met, the session is granted but not secure.

By default, the security modes for encryption and client authentication are both set to soft, which allows unauthenticated and unencrypted access to Sun Ray Clients running older firmware.



Note

Security mode settings don't apply to Oracle Virtual Desktop Clients. Oracle Virtual Desktop Clients will always be treated as if hard security mode for encryption or authentication is in effect.

Table 11.1, "Security Modes" describes what happens when the different security modes are used.

Table 11.1. Security Modes

Situation	Hard Security Mode	Soft Security Mode
Encryption - The Sun Ray Client does not support encryption or server authentication because of old firmware.	Sun Ray server denies the session.	Sun Ray server grants the client a non-secure session. The user must then decide whether to continue using a non-secure session.
Client Authentication - The Sun Ray Client does not support client authentication because of old firmware.	Sun Ray server denies the session.	Sun Ray server grants the client a non-secure session.
Client Authentication - The client supports authentication, but the authentication fails.	Sun Ray server denies the session.	Sun Ray server denies the session.

11.2.2. How to Force Encryption

By default, upstream and downstream encryption is disabled. This procedure provides the steps needed to force upstream and downstream encryption.

Command-Line Steps

- Use the following command to force upstream and downstream encryption:

```
# utcrypto -a enc_up_type=ARCFOUR enc_down_type=ARCFOUR mode=hard
```

Use `-m` instead of `-a` if a non-default security policy already exists.

Admin GUI Steps

1. Navigate to the **Advanced > Security** page.
2. Select the **Upstream Encryption** and **Downstream Encryption** options and select **Hard** as the Security Mode.
3. Click **Save**.

11.2.3. How to Force Server Authentication

By default, server authentication is disabled. This procedure provides the steps needed to force server authentication for all clients.

Command-Line Steps

- Use the following command to force server authentication.

```
# utcrypto -a auth_down_type=simple mode=hard
```

Use `-m` instead of `-a` if a non-default security policy already exists.

Admin GUI Steps

1. Navigate to the **Advanced > Security** page.
2. Select the **Server Authentication** option and select **Hard** as the Security Mode.

3. Click **Save**.

11.2.4. How to Disable Client Authentication

Some reasons to disable client authentication are:

- Reduce administrative overhead: At the cost of security, disabling client authentication saves time required to manage client keys on the servers.
- Eliminate log messages during upgrade: If you upgrade a Sun Ray server in a failover group with older servers, the upgraded server will repeatedly produce log messages indicated that it cannot store key data and the server will treat all keys as unconfirmed. Client authentication should be enabled once the entire group is upgraded.



Note

Disabling client authentication creates a security risk. Make sure you understand the consequences before disabling client authentication.

Before You Begin

- Disabling client authentication applies to all future connections without restarting the Sun Ray server.

Command-Line Steps

- Use the following command to disable client authentication:

```
# utcrypto -a auth_up_type=none
```

Use `-m` instead of `-a` if a non-default security policy already exists.

To enable client authentication, set the `auth_up_type` value to `default`.

Admin GUI Steps

On the **Advanced > Security** page, deselect **Client Authentication** and click **Save**.

11.2.5. How to Force Client Authentication From All Clients

If you don't need to allow access to clients running older versions of firmware, you can improve security by requiring client authentication from all clients.

Command-Line Steps

- Use the following command to force client authentication.

```
# utcrypto -a auth_up_type=DSA auth_mode=hard
```

Use `-m` instead of `-a` if a non-default security policy already exists.

Admin GUI Steps

1. Navigate to the **Advanced > Security** page.
2. Select the **Client Authentication** option and select **Hard** as the Security Mode.
3. Click **Save**.

11.3. Managing Client Keys

A client (a Sun Ray Client or Oracle Virtual Desktop Client) that supports client authentication has a public-private key pair for client authentication. The key pair for a client is generated when the client first boots with the appropriate firmware.



Note

Older versions of firmware or the firmware that is preinstalled on Sun Ray Clients delivered from the factory do not generate keys and do not support client authentication. To help you identify preinstalled firmware, note that versions of preinstalled firmware start with [MfgPkg](#). You must update the firmware on the Sun Ray Clients in order to have keys generated.

When a client connects to a server and client authentication is enabled, the client sends its public key and a client identifier to the server. For a Sun Ray Client, the client identifier is its MAC address. Initially the server can verify only that the client is the owner of the submitted key, but it cannot verify that the client legitimately uses the submitted client ID.

The Sun Ray server stores a list of known clients and their public keys in the Sun Ray data store. A stored key can be marked as confirmed to indicate that authenticity of the key for the given client has been confirmed through human intervention. As long as no key has been marked confirmed for a client, the client authentication feature can ensure only that a client identifier is not used by multiple different clients with different keys. Only when the key has been verified and marked confirmed can the client authentication actually authenticate the identity of the client.



Note

Keys for Oracle Virtual Desktop Clients are not stored in the data store and they are not displayed by the [utkeyadm](#) command or Admin GUI. Instead, an Oracle Virtual Desktop Client uses its key fingerprint as a client identifier so that the authenticity of the key for the given ID is established automatically. For more information, see [Section 11.3.1, “Key Fingerprint”](#).

By default, a client with an unconfirmed key is granted a session unless the identity of the client has been used with a different key. Multiple keys submitted for a client might indicate an attack on sessions for this client, so session access is denied for this client. A user needs to explicitly confirm one of the keys as being authentic to re-enable access for the client.

You can select a stricter policy that requires authenticated client identities and denies access to any client whose key is not verified and confirmed by using the [utpolicy](#) command or the Admin GUI. If you choose to use this policy, you must explicitly mark the key for every new client as 'confirmed' before the client can be used. To use this policy to full effect, you should also set the client authentication mode to 'hard' in the security configuration.

You can use the [utkeyadm](#) command to manage client identities and their associated keys. All keys that are used for a client are listed by the key management tools.

With the [utkeyadm](#) command, you can perform the following actions:

- List keys associated to known clients and their status
- Confirm a client key after verifying its authenticity. If multiple unconfirmed keys are stored for a client, all other keys are deleted when one is confirmed as authentic.
- Delete invalid or stale key entries

- Export key data for all or selected client identities for backup and for transfer to other Sun Ray server instances
- Import key data that has been exported on this or another Sun Ray server instance

You can also view, confirm, or delete associated keys for a client through the client's Desktop Properties page in the Admin GUI.

11.3.1. Key Fingerprint

A key fingerprint is a name for a key and it is what the user can see. A key fingerprint is generated by an MD5 hash based on the public key data.

You can view the key fingerprint for a client in the key panel. To display the key panel, press Stop-K on an Oracle keyboard or Ctrl-Pause-K on a non-Oracle keyboard. To verify the authenticity of a client key, you can compare the key fingerprint displayed in the client's key panel with the one shown by the `utkeyadm` command for the same client.

11.3.2. How to Deny Access to Clients With Unconfirmed Keys

Sun Ray Client keys are initially considered unconfirmed and need to be confirmed as authentic for the specific client by human intervention. Oracle Virtual Desktop Client keys are always considered automatically confirmed (auto-confirmed), because the ID by which a Desktop Access Client is identified is uniquely derived from its key.

The following procedure sets the policy that a confirmed key is required before access to a client is granted. To enact a stronger policy, you should also set up the security policy to require client authentication from all clients, as described in [Section 11.2.5, “How to Force Client Authentication From All Clients”](#).

Command-Line Steps

1. View the current policies:

```
# utpolicy
Current Policy:
-a -g -z both -k pseudo -u pseudo
```

2. Set the client authentication policy with the `-c` option:

```
# utpolicy -a -g -z both -k pseudo -u pseudo -c
```

3. Restart the Sun Ray services:

```
# utstart
```

Admin GUI Steps

1. On the **Advanced > System Policy** tab page, select the **Client Key Confirmation Required** option in the Client Authentication section.
2. Restart all servers in the server group.

11.3.3. How to Confirm a Specific Client Key

This procedure is required if a client receives a Keyerror (49) or Session Refused (50) icon due to conflicting or unconfirmed keys. Once the key is confirmed, you must disconnect the client by rebooting or inserting and removing a smart card to access a session after the change.

Before You Begin

- View the unconfirmed keys (key fingerprints) for all or specific clients.
- To determine whether an unconfirmed client key really belongs to that client, display the key fingerprint for the client by pressing Stop-K.

Command-Line Steps

```
# utkeyadm -a -c IEEE802.000000ee0d6b
1 key confirmed .
# utkeyadm -a -c IEEE802.00000f85f52f -k 1c:d4:b9:31:9d:f0:00:ba:db:ad:65:6c:8e:80:4d:b3
1 key confirmed .
```

Admin GUI Steps

1. Go to the Desktop Unit Properties page for a single client.
2. In the Client Keys table, select a single key and click **Confirm**.

11.3.4. How to Confirm All Unconfirmed Client Keys

If you are certain that all clients requiring key confirmation have been connected to the server group (their genuine keys are stored on the server) and if you are certain that no unwanted clients have keys stored on the server, then you can summarily confirm all known unconfirmed keys. If conflicting keys exist for a client, that client will be skipped.

1. Display all the client keys.

```
# utkeyadm -l -H
```

For example:

```
# utkeyadm -l -H
CID TYPE KEY-FINGERPRINT STATUS
IEEE802.00000adc1a7a DSA* 4f:98:25:60:3b:fe:00:ba:db:ad:56:32:c3:e2:8b:3e confirmed
IEEE802.00000f85f52f DSA* 1c:d4:b9:31:9d:f0:00:ba:db:ad:65:6c:8e:80:4d:b3 unconfirmed
IEEE802.00000f85f52f DSA* 4f:98:25:60:3b:fe:00:ba:db:ad:56:32:c3:e2:8b:3e unconfirmed
IEEE802.00000fe4d445 DSA* 13:d0:d4:47:aa:7f:00:ba:db:ad:26:3a:17:25:11:24 unconfirmed
IEEE802.000000ee0d6b DSA* d0:d7:d0:57:12:18:00:ba:db:ad:b7:0f:5a:c0:8b:13 unconfirmed
```

2. Confirm all unconfirmed client keys.

```
# utkeyadm -a -U
Skipping cid=IEEE802.00000f85f52f: Multiple (2) keys found.
2 keys confirmed.
```

Using the previous example, the unconfirmed client keys for [IEEE802.00000fe4d445](#) and [IEEE802.000000ee0d6b](#) are confirmed.

11.3.5. How to Display a Client's Fingerprint Key from a Sun Ray Client

To display the key fingerprint for a client, press the Stop-K key combination on an Oracle keyboard or Ctrl-Pause-K on a non-Oracle keyboard.

If the key panel does not display, the client might have old firmware installed that doesn't support client authentication.

If the message **No key available** is displayed, the client still has preinstalled **MfgPkg** firmware or a bug exists.

11.3.6. How to Display All Client Keys

This procedure shows how to display client keys in the data store. For additional options to display client keys, see the [utkeyadm](#) man page.

Command Line Steps

- Use the [utkeyadm](#) command.

```
# utkeyadm -l -H
```

For example:

```
# utkeyadm -l -H
CID TYPE KEY-FINGERPRINT STATUS
IEEE802.00000adc1a7a DSA* 4f:98:25:60:3b:fe:00:ba:db:ad:56:32:c3:e2:8b:3e confirmed
IEEE802.00000f85f52f DSA* 1c:d4:b9:31:9d:f0:00:ba:db:ad:65:6c:8e:80:4d:b3 unconfirmed
IEEE802.00000f85f52f DSA* 4f:98:25:60:3b:fe:00:ba:db:ad:56:32:c3:e2:8b:3e unconfirmed
IEEE802.00000fe4d445 DSA* 13:d0:d4:47:aa:7f:00:ba:db:ad:26:3a:17:25:11:24 unconfirmed
IEEE802.000000ee0d6b DSA* d0:d7:d0:57:12:18:00:ba:db:ad:b7:0f:5a:c0:8b:13 unconfirmed
```

Admin GUI Steps

- For multiple clients, click the Desktop Units tab.

The Client Key Status column indicates whether the client has a key in a confirmed or unconfirmed status, whether the client has multiple unconfirmed keys creating a conflict, or whether a key exists for the client. The possible Client Key Status values are None, Unconfirmed, Confirmed, Conflict, Automatic, or Invalid.

11.3.7. How to Display All Keys for a Specific Client

This procedure shows how to display client keys in the data store. For additional options to display client keys, see the [utkeyadm](#) man page.

Command-Line Steps

- Use the [utkeyadm](#) command.

```
# utkeyadm [-l|-L] -c cid -H
```

where [cid](#) is the desktop ID of the client and [-L](#) displays additional auditing information.

Example

The following example displays all keys for the IEEE802.0003ba0d93af client with additional auditing information.

```
# utkeyadm -L -c IEEE802.0003ba0d93af -H
CID TYPE KEY-FINGERPRINT STATUS CREATED CONFIRMED CONFIRMED BY
IEEE802.0003ba0d93af DSA* 4f:98:25:60:3b:fe:d6:f8:fb:38:56:32:c3:e2:8b:3e unconfirmed
2009-06-01 05:08:50 UTC -
```

Admin GUI Steps

- For a single client, go to the Desktop Unit Properties page.

The Client Keys table shows the known keys and their status for the client.

11.3.8. How to Delete a Specific Client Key

- To delete a specific client key, use the following command:

```
# utkeyadm -d -c cid -k key-id
```

where *cid* is the desktop ID of the desktop to which the key belongs and *key-id* is the key fingerprint.

For example:

```
# utkeyadm -d -c IEEE802.00000f85f52f -k 1c:d4:b9:31:9d:f0:00:ba:db:ad:65:6c:8e:80:4d:b3
1 key deleted .
```

11.3.9. How to Delete All Client Keys for a Specific Client

- To delete all client keys for a specific client, type the following command:

```
# utkeyadm -d -c cid
```

where *cid* is the desktop id of the desktop to which the keys belong.

For example:

```
# utkeyadm -d -c IEEE802.00000f85f52f
2 keys deleted.
```

11.4. Displaying Security Status

This section shows how to display the current security status for Sun Ray Clients.

11.4.1. How to Display Security Status for a Sun Ray Client

Once a connection has been successfully established between a client and a server, you can display a client's security status by pressing Stop-N (or Ctrl-Pause-N) to display a security status icon and the Sun Ray Client's MAC.

For a description of OSD icons and their respective codes, see [Chapter 16, Troubleshooting Icons](#).

11.4.2. How to Display Security Status for All Sessions

To display the security status for all sessions on a Sun Ray server, type the following command:

```
# utsession -p
```

Output similar to the following example will be displayed.

```
Token ID Registered Name Unix ID Disp State
Payflex.0000074500000202 ??? ??? 2 IEA
Micropayflex.000003540004545 ??? ??? 3 D
```

The State column displays the encrypted/authenticated state of the session, as shown in [Table 11.2, "utsession State Descriptions"](#)

Table 11.2. utsession State Descriptions

State Column Value	Description
E	Encrypted session

State Column Value	Description
A	Server is authenticated
C	Authenticated client with confirmed identity, including software clients with automatically confirmed keys
U	Authenticated clients with unconfirmed identity. Such connections might not have regular session access if the current policy requires a confirmed identity.
X	Clients that have successfully authenticated with an unconfirmed key, but that key is in conflict with other equally unconfirmed keys that have been used with the same client ID. Clients that have a conflicting key will not be granted session access and you need to confirm one of the known keys as authentic in order to admit the affected clients again.

For more information, see the [utsession](#) man page.



Note

A multihead group might have clients at different firmware levels. The [utsession](#) output shows the lowest security level across the set of all clients participating in the multihead group. For example, if at least one of the clients does not support encryption or authentication, the session will be marked as not encrypted or not authenticated.

11.5. Authentication Troubleshooting

This section provides the possible error messages for client and server authentication.

11.5.1. Error Messages

Errors in authentication are reported in the following log files:

- Installation logs:
 - [/var/adm/log](#) (Oracle Solaris only)
 - [/var/log](#) (Oracle Linux only)
- Configuration logs:
 - [/var/adm/log](#) (Oracle Solaris only)
 - [/var/log/SUNWut](#) (Oracle Linux only)
- General log files:
 - [/var/opt/SUNWut/log](#)
 - [/var/opt/SUNWut/srds/log](#)
 - [/var/opt/SUNWut/srds/replug](#)

Messages logged into [/var/opt/SUNWut/log/messages](#) are delivered through the [syslog](#) service described in the [syslogd](#) man page. The general format of these messages is:

```
timestamp thread_name message_class message
```

For example:

```
May 7 15:01:57 e47c utauthd: [ID 293833 user.info] Worker3 NOTICE: SESSION_OK pseudo.080020f8a5ee
```

Message components are defined as follows:

- *timestamp* format: *year.month.day hours:minutes:seconds*
- *thread_name*:
 - Worker# - Handles client authentication, access control, and session monitoring. Messages with the same thread name are related. The exception occurs when a Worker# thread disconnects a client and then purges the connection information from memory. After a Worker# DESTROY message, the next use of that Worker# thread name has no relation to previous uses of the thread name. In other words, thread names are reused.
 - SessionManager# - Communicates with *utsessiond* on behalf of a Worker# thread.
 - AdminJobQ - Used in the implementation to wrap a library that would not otherwise be thread-safe.
 - CallBack# - Communicates with applications such as *utload*.
 - WatchID - Used to poll data or terminals from connections
 - Terminator - Cleans up terminal sessions
 - Group Manager - Main group manager thread
- *message_class*:
 - CLIENT_ERROR - Indicates unexpected behavior from a client. These messages can be generated during normal operation if a client is rebooted.
 - CONFIG_ERROR - Indicates a system configuration error. The Authentication Manager exits after this error is detected.
 - NOTICE - Indicates a normal event.
 - UNEXPECTED - Logs events or conditions that were not anticipated for normal operation but are not fatal.
 - DEBUG - Occurs only if explicitly enabled and is used by the development team. Debug messages can reveal session IDs, which must be kept secret to ensure proper security.

Table 11.3. Server and Client Authentication Error Message Examples

Error class	Message	Description
CLIENT_ERROR	...Exception ... : cannot send keepAliveInf	Error encountered while attempting to send a keep-alive message to a client.
	...keepAlive timeout	A client has failed to respond within the allotted time. The session is being disconnected.
	duplicate key:	Client does not properly implement the authentication protocol.
	invalid key:	Client does not properly implement the authentication protocol.

Error Messages

Error class	Message	Description
CONFIG_ERROR	attempt to instantiate CallBack 2nd time.	Program error.
	AuthModule.load	Problem encountered while loading configuration module.
	Cannot find module	Program or installation error.
NOTICE	"discarding response: " + param	No controlling application is present to receive client response.
	"NOT_CLAIMED PARAMETERS: " + param	A token was not claimed by any authentication module.
	...authentication module(s) loaded.	Notification that authentication modules have loaded.
	...DISCONNECT ...	Normal notification of disconnection.
UNEXPECTED	"CallBack: malformed command"	Bad syntax from a user application such as utload or utidle .
	.../ ... read/0:" + ie	Possible program error.
	.../ ... read/1: ... Exception ...	Error encountered while reading messages from the client.
	.../... protocolError: ...	Various protocol violations are reported with this message. This error condition is also a way for utauthd to force the client to reset.

Chapter 12. Multiple Monitor Configurations

Table of Contents

12.1. Multi-Monitor	135
12.1.1. How to Set a Sun Ray Client's Multi-Monitor Configuration With Optimal Settings	136
12.1.2. How to Set a Sun Ray Client's Multi-Monitor Configuration With Customized Settings	136
12.2. Multihead Groups	138
12.2.1. Creating a Multihead Group	138
12.2.2. Multihead Group Screen Indicator	139
12.2.3. Creating a Single Screen Across Several Monitors (Xinerama)	139
12.2.4. How to Create a New Multihead Group	139
12.2.5. How to Enable the Multihead Group Policy	140
12.2.6. How to Manually Set Multihead Group Screen Dimensions	141
12.2.7. How to Manually Set Multihead Group Geometry	141
12.2.8. How to Disable Multihead Group for a Session	142
12.2.9. How to Enable and Disable Xinerama	142
12.2.10. How to Disconnect a Secondary Client	142

This chapter describes how to administer two different types of monitor configurations that Sun Ray Software supports: multi-monitor and multihead groups.

A multi-monitor configuration supports multiple monitors connected to the dual video connectors on a Sun Ray 2FS or Sun Ray 3 Plus Client.

A multihead group configuration enables you to merge and control multiple Sun Ray Clients, referred to in this context as heads, and their screens using a single keyboard and mouse attached to a primary client.

12.1. Multi-Monitor

This section describes the multi-monitor support for the Sun Ray 2FS and Sun Ray 3 Plus Clients with dual video connectors. [Figure 12.1, "Multi-monitor Example"](#) shows a Sun Ray 3 Plus Client using the multi-monitor feature and the Windows connector.

Figure 12.1. Multi-monitor Example



Multi-monitor support for Sun Ray Clients is provided by the X Resize, Rotate, and Reflect (RandR) 1.2 extension, which provides a way to use the multiple monitors as one screen. Features include:

- Configuration changes can be applied dynamically to a session.
- Application windows are aware of monitor boundaries to avoid placement issues.
- No size restrictions. Hotdesking is supported to other clients with different monitor resolutions, whether they are smaller or larger.

RandR 1.2 support is provided with the default Xserver, Xnewt, which is automatically installed and configured with the Sun Ray Software. The optimal multi-monitor configuration is applied automatically when a Sun Ray Client sessions starts, unless the `utxconfig -r` command is used to set the DIMENSIONS parameter. In that case, the `utxconfig` value will be used to size the screen.

A session's multi-monitor configuration is preserved after hotdesking, which means hotdesking between clients with different monitor configurations can produce unacceptable results. The dynamic session resizing feature solves this problem and automatically reconfigures the session's display based on the new screen configuration. See [Section 13.1.2, “Dynamic Session Resizing”](#) for details.

If dynamic session resizing is not enabled, then you have two ways to update the new display configuration after hotdesking, either automatically with optimal settings or manually with customized settings. See [Section 12.1.1, “How to Set a Sun Ray Client's Multi-Monitor Configuration With Optimal Settings”](#) and [Section 12.1.2, “How to Set a Sun Ray Client's Multi-Monitor Configuration With Customized Settings”](#) for details.



Note

The gnome-display-properties GUI monitor configuration tool should not be used on Oracle Solaris or Oracle Linux. Using this tool may adversely affect the client's RandR 1.2 configuration.

12.1.1. How to Set a Sun Ray Client's Multi-Monitor Configuration With Optimal Settings

The following command sets a Sun Ray Client's multi-monitor configuration using the optimal settings provided by `xrandr`. This command is not needed if dynamic session resizing is enabled.

```
/opt/SUNWut/bin/utscreenresize -s all
```

12.1.2. How to Set a Sun Ray Client's Multi-Monitor Configuration With Customized Settings

There may be times when you want to create a specific multi-monitor configuration for a Sun Ray Client. This example shows how to use the `xrandr` command to view and set a specific multi-monitor configuration. For more details, refer to the `xrandr` man page. This procedure will not work if dynamic session resizing is enabled.



Note

Sun Ray Software provides an updated version of the `xrandr` command for Oracle Linux, and it is installed in the `/opt/SUNWut/bin` directory during the installation process. The `/opt/SUNWut/bin/xrandr` command provides the required functionality for the Sun Ray Software environment.

Issuing the `xrandr` command without any options shows a client's current multi-monitor configuration. For example, here is the `xrandr` output for a session created on a client with one monitor displaying at 1280x1024 resolution:

```
(use /opt/SUNWut/bin/xrandr on Oracle Linux)
% xrandr
Screen 0: minimum 640 x 480, current 1280 x 1024, maximum 10240 x 10240
DVI1 connected 1280x1024+0+0 (normal left inverted right) 361mm x 288mm
  1280x1024    59.9*+   74.9    65.9
  1152x900     65.8
  1024x768     74.9    69.8    59.9
  800x600      59.9
  640x480      59.4
```

In this output, the Screen line provides the current overall screen resolution (1280x1024) and the available maximum resolution (10240x10240). There is a single Output named DVI1 that shows a 1280x1024 monitor connected to the client's DVI port (or the first DVI port on a dual-DVI client). All resolutions available for this Output are listed. The current mode is indicated by a '*' and the optimal mode indicated with a '+'.

After hotdesking to a dual-monitor client, the client's Screen configuration does not change, but the RandR information is updated to reflect optimal modes. For example, hotdesking to a client with 1600x1200 and 1920x1200 monitors would show the following `xrandr` output:

```
(use /opt/SUNWut/bin/xrandr on Oracle Linux)
% xrandr
Screen 0: minimum 640 x 480, current 1280 x 1024, maximum 10240 x 10240
DVI1 connected 1280x1024+0+0 (normal left inverted right) 451mm x 338mm
  1600x1200    59.9 +
  1280x1024    74.9*   65.9    59.9
  1152x900     75.8    75.0    65.8
  1024x768     74.9    59.9
  800x600      59.9
  640x480      59.4
DVI2 connected (normal left inverted right)
  1920x1200    59.9 +
  1600x1200    59.9
  1280x1024    74.9    65.9    59.9
  1152x900     75.8    75.0    65.8
  1024x768     74.9    59.9
  800x600      59.9
  640x480      59.4
```

For this new client, two Outputs are listed, DVI1 and DVI2. Although DVI2 is "connected," it is not configured with a current mode. Both Outputs have their optimal modes indicated with a '+', but DVI1 still has 1280x1024 as its current mode.

The best way to reconfigure this Screen with the optimal modes for both DVI1 and DVI2 would be to use the `utscresize` command described in the previous section. However, if you wanted to select specific resolutions for each monitor, you would need to use the `--output` option of the `xrandr` command. For example, if you wanted both monitors to use 1600x1200, you would issue the following `xrandr` command:

```
(use /opt/SUNWut/bin/xrandr on Oracle Linux)
% xrandr --output DVI1 --mode 1600x1200 --output DVI2 --mode 1600x1200 --right-of DVI1
```

Here is the new multi-monitor configuration, with both DVI1 and DVI2 at 1600x1200 resolution and DVI2 starting at 1600,0 on the screen:

```
(use /opt/SUNWut/bin/xrandr on Oracle Linux)
```

```
% xrandr
Screen 0: minimum 640 x 480, current 3200 x 1200, maximum 10240 x 10240
DVI1 connected 1600x1200+0+0 (normal left inverted right) 451mm x 338mm
  1600x1200    59.9*+
  1280x1024    74.9    65.9    59.9
  1152x900     75.8    75.0    65.8
  1024x768     74.9    59.9
  800x600      59.9
  640x480      59.4
DVI2 connected 1600x1200+0+0 (normal left inverted right) 519mm x 324mm
  1920x1200    59.9 +
  1600x1200    59.9*
  1280x1024    74.9    65.9    59.9
  1152x900     75.8    75.0    65.8
  1024x768     74.9    59.9
  800x600      59.9
  640x480      59.4
```

Note that 1920x1200 is still marked as "optimal" for DVI2, but it is currently using 1600x1200.

12.2. Multihead Groups

A multihead group configuration enables you to merge and control multiple Sun Ray Clients, referred to in this context as heads, and their screens using a single keyboard and mouse attached to a primary client. The maximum number of clients that can be connected to a single Sun Ray session as a multihead group is 16, and a multihead group can be composed of virtually any mix of Sun Ray Clients. If you use clients that support up to two monitors, such as the Sun Ray 2FS Client or Sun Ray 3 Plus Client, then a multihead group can control as many as 32 monitors. Each Sun Ray Client presents an X screen of the multihead X display.

All peripherals are attached to the primary client. The remaining clients, called the secondary clients, provide the additional displays. If a secondary client tries to connect when the primary client is not available, a Waiting for Primary icon is displayed on the secondary client until the primary client is discovered.

Ordinarily, a multihead group session has a separate desktop toolbar and separate workspaces on each monitor. Windows cannot be moved from one monitor to another within the multihead group.

Xinerama, by contrast, enables all the monitors in a multihead group to be treated as a single screen. See [Section 12.2.9, "How to Enable and Disable Xinerama"](#) for more details.

If you hotdesk from a multihead group to a Sun Ray Client that is not part of a multihead group, you can still view all the screens created in the original multihead group on the single screen by panning to each screen in turn. This action is called *screen flipping*.

12.2.1. Creating a Multihead Group

There are two main steps to create a multihead group:

- Create the multihead group using the `utmhconfig` (GUI) or `utmhadm` command. The command must be run on the primary client. You can use a smart card to identify the terminals.
- Enable the multihead policy using either the `utpolicy` command or the Admin GUI.

Note the following limitations:

- The Sun Ray 2FS and Sun Ray 3 Plus Clients are designed to run a single display across two monitors without additional configuration, using a single frame buffer for two monitors, always treating them as a

single, unified display surface to be controlled with a single mouse and keyboard and displayed to the X server as a single screen.

- Video acceleration works only on the primary client. In a multihead group, the audio stream is directed only to the primary client, so audio/video synchronization is provided only on the primary client.
- Regional hotdesking is not enabled for multihead groups.

12.2.2. Multihead Group Screen Indicator

When the multihead feature is used, a small window indicating the current session on each screen is displayed, with the current screen highlighted for easy identification. This window is automatically displayed for users during session creation. For example, [Figure 12.2, “Multihead Group Screen Indicator”](#) indicates that the user is on the second screen of a three-screen display.

Figure 12.2. Multihead Group Screen Indicator



12.2.3. Creating a Single Screen Across Several Monitors (Xinerama)

The Xinerama extension to X11 creates a single large screen displayed across several monitors. With Xinerama, only one toolbar is displayed, and a window can be moved smoothly from one part of the screen to the next. When Xinerama is enabled, the RandR extension is automatically disabled.

For CDE desktop sessions, a single CDE toolbar and set of workspaces manages the configured monitors. A window including the CDE toolbar itself can span monitors, because the monitor displays are still within the same screen.

See [Section 12.2.9, “How to Enable and Disable Xinerama”](#) for more details.

Xinerama can also be used with the Windows connector `uttsc` command.

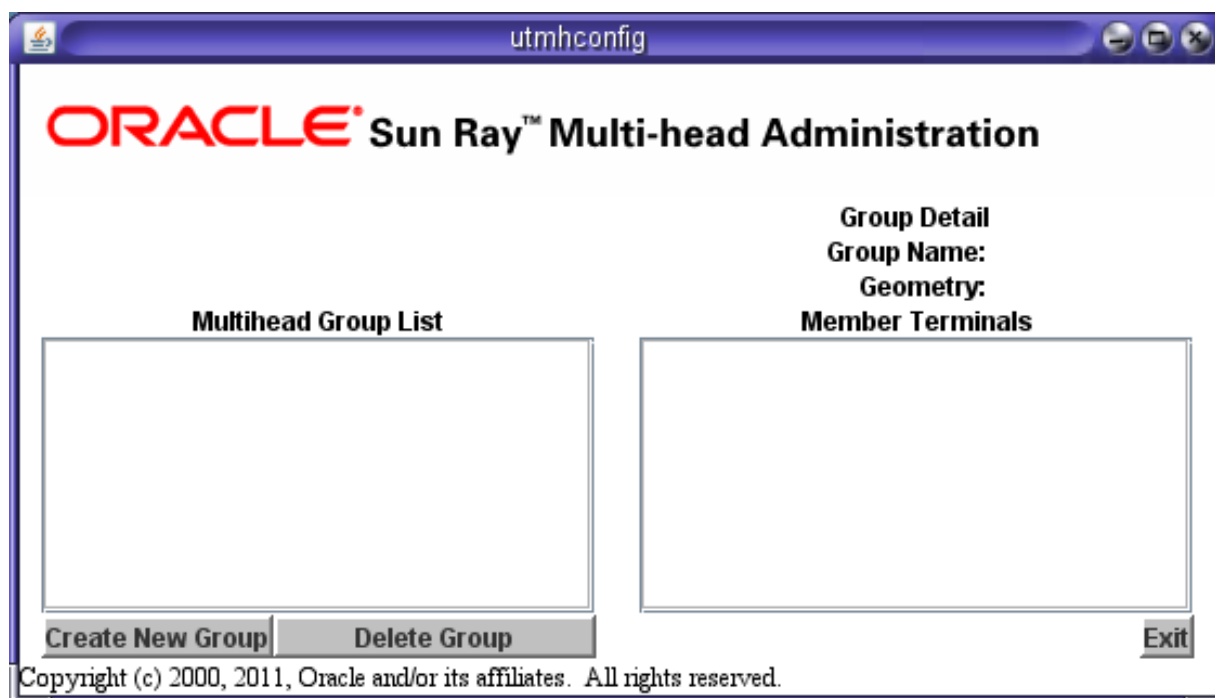
12.2.4. How to Create a New Multihead Group

1. On the primary client, start the Multi-head Administration GUI.

```
# /opt/SUNWut/sbin/utmhconfig
```

2. On the initial screen, click Create New Group.

Figure 12.3. utmhconfig Home Screen



The Create New Multiheaded Group dialog box is displayed. The number of rows and the number of columns you provide are displayed as the group geometry when the group is created.

3. Follow the instructions in the wizard to complete the procedure.

The main step in the wizard is to select the clients within the multihead group and insert a smart card in each Sun Ray Client in turn to establish the order of the group.

4. Click the Finish button.
5. Exit the session or disconnect by removing your card.
6. Enable the Multihead policy.

See [Section 12.2.5, “How to Enable the Multihead Group Policy”](#) for more details.

12.2.5. How to Enable the Multihead Group Policy

Command-Line Steps

The following command enables the multihead group policy for the failover group and restarts the Sun Ray Software with the new policy on the local server without disrupting existing sessions.

```
# /opt/SUNWut/sbin/utpolicy -a -m -g your_policy_flags
# /opt/SUNWut/sbin/utstart
```



Note

Issue the `utstart` command on every server in the failover group.

Admin GUI Steps

1. Click the Advanced tab.
2. Click the System Policy tab.
3. Select the **Multihead Feature Enabled** option.
4. Click **Save**.

If a system restart is needed, an advisory message will display.

12.2.6. How to Manually Set Multihead Group Screen Dimensions

Screen dimensions for the multihead group are automatically set, by default, to the largest dimension supported by the primary client.

To override the automatic sizing of screen dimensions, use the `utxconfig -r` command.



Note

If explicit screen dimensions are chosen, or if the resolutions of the monitors differ, you may have problems with unwanted on-screen movement called panning, or large black bands around the visible screen area.

12.2.6.1. How to Override Automatic Sizing of Screen Dimensions

```
% utxconfig -r widthxheight
```

For example:

```
% utxconfig -r 1280x1024
```

12.2.6.2. How to Restore Automatic Sizing Behavior on the Next Login

```
% utxconfig -r auto
```

12.2.7. How to Manually Set Multihead Group Geometry

A multihead group can have its screens arranged in various configurations. For example, a user can arrange a multihead group of four screens as two rows of two screens (2x2) or as a single row of four screens (4x1). By default, when a user logs into a multihead group, the session uses the number of screens available. The layout or geometry of these displays is generated automatically.

When the mouse pointer is moved past the edge between two screens, it moves from one screen to the next. The geometry of the multihead group determines which screen is displayed at that moment.

You can use the `utxconfig -R` command to manipulate the automatic geometry.

12.2.7.1. How to Override the Automatic Geometry

```
% utxconfig -R columnsxrows
```

12.2.7.2. How to Restore the Automatic Geometry on the Next Login

```
% utxconfig -R auto
```

12.2.8. How to Disable Multihead Group for a Session

```
% utxconfig -m off
```

12.2.9. How to Enable and Disable Xinerama

Users can enable or disable Xinerama as part of their X preferences. The `utxconfig` command handles this setting on an individual token basis. The user must log off for the changes to take effect. When Xinerama is enabled, the RandR extension is automatically disabled.



Note

Video acceleration is not available for Xinerama sessions. Video played without video acceleration can be dragged from one Sun Ray Client to another or can span multiple clients.



Note

Xinerama tends to consume noticeable amounts of CPU, memory, and network bandwidth. For optimal performance, set the `shmsys:shminfo_shmmax` parameter in the `/etc/system` file to at least `LARGEST_NUMBER_OF_HEADS * width * height * 4`.

12.2.9.1. How to Enable Xinerama

1. Log in to the Sun Ray session.
2. Enable Xinerama.

```
% utxconfig -x on
```

3. Log off the session and log back in for the changes to take effect.

12.2.9.2. How to Disable Xinerama

1. Log in to the Sun Ray session.
2. Disable Xinerama.

```
% utxconfig -x off
```

3. Log off the session and log back in for the changes to take effect.

12.2.9.3. How to Enable Xinerama as Default for All Clients

1. Become superuser on the Sun Ray server.
2. Enable Xinerama for all clients.

```
# utxconfig -a -x on
```

If users are currently logged in, they need to log off the session and log back in for the changes to take effect.

12.2.10. How to Disconnect a Secondary Client

If you disconnect the secondary clients without deleting the multihead group to which they belong, the screens are not displayed on the single primary client. The primary client is still part of the multihead group, and the mouse cursor can appear to get lost when it goes to the disconnected secondary client.

To recover from this situation, you can do one of the following actions:

1. Reconnect the missing client.
2. Delete the multihead group using the `utmhconfig` or `utmhadm` command.
3. Replace the missing client.
4. Create a new multihead group that incorporates the replacement client.

Chapter 13. Desktop Clients

Table of Contents

13.1. Managing Desktop Clients	146
13.1.1. Client ID Differences Between Oracle Virtual Desktop Clients and Sun Ray Clients	146
13.1.2. Dynamic Session Resizing	147
13.1.3. How to List Available Sun Ray Servers	150
13.1.4. How to List the Available Clients	150
13.1.5. How to Display Sun Ray Client Information	150
13.1.6. How to Configure a Client's Location and Information	151
13.1.7. Audio Output Troubleshooting (Oracle Solaris 10 and Oracle Linux 5)	151
13.1.8. Audio Output Troubleshooting (Oracle Solaris 11 and Oracle Linux 6)	153
13.2. Sun Ray Clients	155
13.2.1. How to Centralize Sun Ray Client Configurations (.parms)	155
13.2.2. Sun Ray Client Hot Keys	158
13.2.3. How to Change Sun Ray Client Audio and Display Settings (Sun Ray Settings GUI)	160
13.2.4. How to Modify Screen Resolutions	161
13.2.5. How to Power Cycle a Sun Ray Client	162
13.2.6. How to Enable or Disable XRender	162
13.2.7. How to Configure Screen Rotation	163
13.2.8. How to Disable Screen Blanking on a Sun Ray Client	163
13.2.9. How to Enable the NumLock Key for All Sun Ray Sessions	165
13.2.10. Keyboard Country Codes	166
13.2.11. Sun Ray Client Boot Process	167
13.3. Oracle Virtual Desktop Clients	170
13.3.1. Oracle Virtual Desktop Clients Overview	170
13.3.2. Using External Devices on the Client Computer	171
13.3.3. How to Enable Access for Oracle Virtual Desktop Clients	171
13.3.4. How to Enable the Clipboard Service for Oracle Virtual Desktop Clients	173
13.3.5. Oracle Virtual Desktop Client Troubleshooting	173

This chapter provides information about the desktop clients supported by Sun Ray Software, which includes Sun Ray Clients and Oracle Virtual Desktop Clients. Each desktop client has its own chapter detailing specific information. This chapter specifies topics that pertain to both types of clients.

Refer to the following sections and chapters for more specific information about the desktop clients.

Sun Ray Clients

- [Section 13.2, "Sun Ray Clients"](#)
- [Chapter 14, *Sun Ray Client Firmware*](#)
- [Chapter 15, *Peripherals*](#)
- [Chapter 16, *Troubleshooting Icons*](#)

Oracle Virtual Desktop Clients

- [Section 13.3, "Oracle Virtual Desktop Clients"](#)
- [Chapter 15, *Peripherals*](#)

- [Chapter 16, Troubleshooting Icons](#)

See the Oracle Virtual Desktop Client documentation (<http://www.oracle.com/technetwork/server-storage/sunrayproducts/docs/index.html>) for detailed information about the Oracle Virtual Desktop Client application and its command-line equivalent, the `ovdc` command.

13.1. Managing Desktop Clients

This section provides the list of common management tasks or feature information for both Sun Ray Clients and Oracle Virtual Desktop Clients. For information specific to the respective clients, see [Section 13.2, “Sun Ray Clients”](#) and [Section 13.3, “Oracle Virtual Desktop Clients”](#).

13.1.1. Client ID Differences Between Oracle Virtual Desktop Clients and Sun Ray Clients

If you have existing scripts using the Sun Ray Software commands, or you plan to create scripts, you must be aware of the client ID differences between Oracle Virtual Desktop Clients and Sun Ray Clients.

All clients are represented in the Sun Ray Software administration tools by a client ID, also called CID, terminal CID, or client identifier. A client ID has both a full ID and a short ID version:

- Full client ID: `namespace.id-part`
- Short client ID: `id-part`

The `namespace` value is a tag that determines the format of the `id-part` value. Short client IDs are usually used and accepted because the current namespaces, one for Sun Ray Clients and one for Oracle Virtual Desktop Clients, use different `id-part` formats. The full client ID is used to help distinguish between these different types of clients more easily.

See [Table 13.1, “Oracle Virtual Desktop Client ID Details”](#) for the details of the client ID.

Table 13.1. Oracle Virtual Desktop Client ID Details

Client	<code>namespace</code> Value	<code>id-part</code> Meaning	<code>id-part</code> Format
Sun Ray Client	IEEE802	MAC address of Sun Ray Client	12 hex digits
Oracle Virtual Desktop Client	MD5	MD5 hash of client key	32 hex digits



Note

The client key is part of an Oracle Virtual Desktop Client profile, so every Oracle Virtual Desktop Client profile has its own client ID.

See [Table 13.2, “Example Sun Ray Client IDs”](#) and [Table 13.3, “Example Oracle Virtual Desktop Client IDs”](#) for examples of client IDs.

Table 13.2. Example Sun Ray Client IDs

Short ID	Full CID
0003badc1b9d	IEEE802.0003badc1b9d
00144f85f52f	IEEE802.00144f85f52f
080020b5ca55	IEEE802.080020b5ca55

Table 13.3. Example Oracle Virtual Desktop Client IDs

Short ID	Full CID
1bd97b44ea9458fac256a7a778a282fe	MD5.1bd97b44ea9458fac256a7a778a282fe
d8b3a4eb29497e0c6fbb0f2a810267f5	MD5.d8b3a4eb29497e0c6fbb0f2a810267f5

13.1.1.1. How to Display Client ID Information

The format of the client ID for an Oracle Virtual Desktop Client is different to the client ID for a Sun Ray Client. See [Section 13.1.1, “Client ID Differences Between Oracle Virtual Desktop Clients and Sun Ray Clients”](#) for more information.

You can display a Sun Ray Client's short ID in the following ways:

- **Oracle Keyboard** - Stop-N
- **Non-Oracle Keyboard** - Ctrl-Pause-N

You can display an Oracle Virtual Desktop Client's short ID in the following ways:

- **Keyboard** - Host-N (By default, Host is the right Ctrl key.)
- **Command** - Use the `-i` or `--clientid` command option of the `ovdc` command.

13.1.2. Dynamic Session Resizing

Dynamic session resizing allows the remote desktop to be resized automatically to fit the optimized size of your local desktop client session. When you hotdesk to a session from a different device, or use a client device like a tablet, which can be rotated, the new screen configuration is detected and the session screen dimensions are adapted accordingly.. See [Section 13.1.2.1, “Resizing Scenarios”](#) for a list of scenarios when dynamic session resizing is used.

With dynamic session resizing, any change in a desktop client's screen configuration is detected and the remote desktop is automatically changed accordingly. Changes to the desktop client's screen configuration can include the number of monitors, monitor resolutions, monitor orientations, tablet rotation, or screen mode, such as window mode or full screen mode. These changes can happen at any time during a session, for example, even when a tablet changes orientation from landscape to portrait.

To enable dynamic session resizing for Sun Ray sessions, you must use one of the following methods to invoke the `-l` option of the `utsscreenresize` command, which runs `utsscreenresize` in a session's background:

- **GNOME autostart** - For users that use regular Oracle Solaris or Oracle Linux sessions, use the GNOME autostart configuration to invoke the `utsscreenresize -l` command when the user's GNOME desktop is initialized. Since this method uses the standard GNOME autostart configuration, users will have the ability to disable it through their GNOME desktop preferences. See [Section 13.1.2.2, “How to Enable Dynamic Session Resizing Using GNOME Autostart”](#) for details.
- **Session initialization script** - For all other non-regular sessions, such as kiosk sessions, use the session initialization script to invoke the `utsscreenresize -l` command when a session initializes. See [Section 13.1.2.3, “How to Enable Dynamic Session Resizing Using a Session Initialization Script”](#) for details.

For Oracle Virtual Desktop Clients, users can also enable or disable this feature on the client side through the configuration settings, and it is enabled by default.

If dynamic session resizing is enabled for a session, all other display resizing operations are ignored, such as the `utxconfig -r` or `xrandr` commands. If users want to configure their own specific screen

configuration or desktop resolutions, you must disable dynamic session resizing for those users so they can use other commands and tools to manage their screen or desktop configurations. Or, in the case of the GNOME autostart configuration, users can disable it themselves.

Dynamic session resizing is available for all Sun Ray Clients, Oracle Virtual Desktop Client, Oracle Virtual Desktop Client for iPad 1.2 or later, and Oracle Virtual Desktop Client for Android 1.2 or later. Oracle Virtual Desktop Client 3.2 or later is required to provide users an optimal display when resizing the window while in window mode.



Note

Dynamic session resizing will not work if the Xinerama extension is enabled. See [Section 12.2.9, “How to Enable and Disable Xinerama”](#) for more details.

13.1.2.1. Resizing Scenarios

Dynamic session resizing will automatically resize the remote desktop in the following situations:

- Hotdesking from a Sun Ray Client to another Sun Ray Client with a different monitor configuration, monitor resolution, or monitor orientation.
- Hotdesking from a Sun Ray Client to an Oracle Virtual Desktop Client running on a client computer, and vice versa.
- Resizing the window while running Oracle Virtual Desktop Client in window mode. The session's display automatically resizes to the new window size. If you reduce the size of a window without desktop session resizing, you must use scroll bars to view the entire display, because the session's display does not automatically resize and is too big to fit in the smaller window.
- Rotating a tablet from landscape to portrait mode while running the Oracle Virtual Desktop Client application, and vice versa. The remote desktop automatically rotates with the tablet.

13.1.2.2. How to Enable Dynamic Session Resizing Using GNOME Autostart

This procedure shows how to enable dynamic session resizing for all users when their GNOME desktop starts. This method is best for regular Oracle Solaris and Oracle Linux sessions.

1. Become superuser on the Sun Ray server.
2. (Sun Ray Clients only) If there are any Sun Ray Clients that have monitors in a non-standard orientation, set the Sun Ray Client's firmware with the valid orientation values ([orient1](#) and [orient2](#)) for the connected monitors.

Use the `-e` option of the `utfwadm` command to update the firmware for a specific Sun Ray Client. See [Section 13.2.1, “How to Centralize Sun Ray Client Configurations \(.parms\)”](#) for details.

3. Add the `utscreeenresize` command with the `-s all -l` option to the GNOME autostart directory.

For Oracle Solaris 11, Oracle Linux 6, and Oracle Linux 5

Add the following file named `utscreeen.resize` to the `/usr/share/gnome/autostart` directory.

```
[Desktop Entry]
Type=Application
Exec=/opt/SUNWut/bin/utscreeenresize -s all -l
Hidden=false
X-GNOME-Autostart-enabled=true
Name[en_US]=Sun Ray Session Screen Resize
Name=Sun Ray Session Screen Resize
Comment[en_US]=Start utscreeenresize on login.
```

```
Comment=Start utoolsresize on login.
```

With this file in place, all users will have dynamic session resizing enabled by default. Individual users may disable this feature by choosing **System > Preferences > Startup Applications** and unchecking the **Sun Ray Screen Resize** application. The user will have to log out and log in for the change to have effect.

If dynamic session resizing is not enabled for all users through `/usr/share/gnome/autostart`, individual users can still enable this feature by creating a Startup Application for the `utoolsresize -s all -l` command. They can choose **System > Preferences > Startup Applications** and click **Add** in the Startup Applications Properties window.

For Oracle Solaris 10

To enable dynamic session resizing, individual users must create a Startup Program for the `utoolsresize -s all -l` command. Click **Launch > Preferences > Desktop Preferences > Sessions**, click the **Startup Programs** tab, and click **Add**.

13.1.2.3. How to Enable Dynamic Session Resizing Using a Session Initialization Script

This procedure shows how to enable dynamic session resizing when the user's session initializes. This method is required for non-regular sessions, such as kiosk sessions.

1. Become superuser on the Sun Ray server.
2. (Sun Ray Clients only) If there are any Sun Ray Clients that have monitors in a non-standard orientation, set the Sun Ray Client's firmware with the valid orientation values (`orient1` and `orient2`) for the connected monitors.

Use the `-e` option of the `utfwadm` command to update the firmware for a specific Sun Ray Client. See [Section 13.2.1, "How to Centralize Sun Ray Client Configurations \(.parms\)"](#) for details.

3. Change directory to the session initialization directory.

- Oracle Solaris 10:

```
# cd /usr/dt/config/Xsession.d
```

- Oracle Linux or Oracle Solaris 11:

```
# cd /etc/X11/xinit/xinitrc.d
```

4. Create the following customized script to enable dynamic session resizing (the script is called `0050.desktopresize.sh` in this procedure).

```
#!/bin/sh

# Enable dynamic session resizing each time a user hotdesks
/opt/SUNWut/bin/utoolsresize -s all -l &
```



Note

The script name should have the `0050.` prefix to make sure it is run at the appropriate time. For Oracle Linux and Oracle Solaris 11, the script name must have the `.sh` extension, otherwise the script will not get sourced.

5. Save the script and make the script executable for everyone.

```
# chmod 775 0050.desktopresize.sh
```

6. Start a new session, so the script gets sourced.

13.1.3. How to List Available Sun Ray Servers

- In a shell window on the client, type the following command:

```
% utswitch -l
```

The available Sun Ray servers to the client within the current server group are displayed.

13.1.4. How to List the Available Clients

This procedure describes how to list all the available Sun Ray Clients and Oracle Virtual Desktop Clients on the Sun Ray server.

Command-Line Steps

1. Become superuser on the Sun Ray server.
2. Display all the available clients.

```
# utdesktop -l
```

Admin GUI Steps

1. Start the Admin GUI.
2. Click the Desktop Units tab.

The list of available clients are displayed. You can use the drop-down menu and **Search** field to display the specific clients you want to view.

13.1.5. How to Display Sun Ray Client Information

This procedure describes how to view detailed information about registered desktop clients, including their client ID. If you have access to the physical Sun Ray Client, you can press Stop-V to view the client's current information.

Command-Line Steps

1. Become superuser on the Sun Ray server.
2. Display information about a client.

```
# utdesktop -p clientID
```

where *clientID* is the short ID of the client, as described in [Section 13.1.1, "Client ID Differences Between Oracle Virtual Desktop Clients and Sun Ray Clients"](#). You can use the `utdesktop -l` command to list all the desktop clients and their client IDs.

Admin GUI Steps



Note

To facilitate the searching process, you can use the Admin GUI to edit desktop client properties. Click the DTU Identifier and then click **Edit**. You can then provide a location or other information.

1. Start the Admin GUI.

2. Click the Desktop Units tab.
3. From the Desktop Units tab, choose the information to display:
 - To display information about a specific desktop client, click on the DTU Identifier (MAC address) or enter a search string in the text field.
 - To display information about a group of desktop clients, select an option from the drop-down menu (All Connected Desktop Units, Token Readers, or Multihead Groups) and/or enter a search string in the text field to narrow your search.

13.1.6. How to Configure a Client's Location and Information

This procedure enables you to add the location and other information about a desktop client. This is useful for various applications like the location awareness feature.

Command-Line Steps

- Edit the client location and other client information for a Sun Ray Client:

```
# utdesktop -e "client_id,location,[other_info]"
```

- Edit the client location and other client information for a batch of Sun Ray Clients:

```
# utdesktop -e -f filename
```

filename must be a comma-separated format (CSV) file.

- List the configured client location and other client information for all Sun Ray Clients:

```
# utdesktop -l
```

Admin GUI Steps

1. Click the Desktop Units tab.
2. Select a client ID to display its Desktop Unit Properties screen.
3. Click **Edit** to display the Edit Desktop Unit Properties screen.
4. Enter a location and other information about the client.
5. Click **Save**.

13.1.7. Audio Output Troubleshooting (Oracle Solaris 10 and Oracle Linux 5)

This section provides troubleshooting information for audio output on Sun Ray Clients and Oracle Virtual Desktop Clients running Oracle Solaris 10 or Oracle Linux 5 sessions.

13.1.7.1. Audio Frequencies Used With Applications

A desktop client uses whatever audio frequency an application needs, which enables you to configure application audio requirements to help reduce bandwidth and increase scalability. For example, if a VoIP application requests 8kHz mono, a desktop client will transmit only 8kHz mono.

13.1.7.2. Tracking Audio Sessions

Each time a user logs in to a desktop client, a script automatically assigns the `$AUDIODEV` environment variable to that session. One `utaudio` process is assigned to each session. Refer to the `utaudio` and `audio` man pages for more information.

13.1.7.3. Audio Device Emulation During Hotdesking

During hotdesking, an emulated audio device follows the user to the new session. The name of the emulated device is carried in the `$AUDIODEV` environment variable. It is the responsibility of the audio application to inspect `$AUDIODEV` and direct its output to that device.

The emulated audio devices are created as device nodes in the `/tmp/SUNWut/dev/utaudio` directory. This directory tree is recreated at boot time.



Note

Do not remove the `/tmp/SUNWut/dev/utaudio` directory. If you delete this directory, users with `utaudio` sessions cannot use their audio pseudo device nodes.

13.1.7.4. Problem: Audio is not working

- Use the Oracle keyboard audio keys (top right of keyboard) and check the volume and mute buttons.
- Display the Sun Ray session's audio settings:

```
$ utsettings
```

and verify that the audio output is selected properly, for example, for headphones or speakers.

- Make sure the volume is not muted in your desktop session.
- Try a set of external speakers plugged into the Sun Ray Client's audio out or headphones port. If that works, the Sun Ray Client might have a broken speaker.
- To test whether the audio is working, type the following:

```
$ cat audiofile > $AUDIODEV
```

Oracle Solaris provides suitable sample PCM-encoded audio files in `/usr/share/audio/samples/au`, so for instance this command:

```
$ cat /usr/share/audio/samples/au/gong.au > $AUDIODEV
```

should produce the sound of a gong.

Linux generally does not provide PCM-encoded audio files. If you can not locate a suitable file then this command can be used to generate a continuous tone:

```
$ perl -e 'foreach(-8..8){push(@v,pack("n",4*$_))} while(1){print @v}' > $AUDIODEV
```

If the `cat` or `perl` command hangs, you might need to quit any other applications that are currently trying to play audio, for example, a browser.

13.1.7.5. Problem: Audio is not working with Firefox

- Check the current release of the Flash plugin and make sure it is at version 9.0.r125 or later. To check the Flash plugin version, type `about:plugins` as the URL in the browser.
- Try quitting Firefox and explicitly restart it in a terminal window.
- If all else fails, quit Firefox, go to your `.mozilla` directory, and rename the "firefox" directory to something else, for example, `firefox.jan09`. Then, restart Firefox and see whether the audio works with a completely clean configuration.

If the audio works with the clean configuration, then something is wrong in your browser's previous configuration.

13.1.7.6. Problem: Audio is not working with latest versions of Firefox or Adobe Flash Player (Oracle Linux 5)

This problem is likely occurring because you are using the latest releases of Firefox or Adobe Flash Player, which provide only ALSA sound support. Oracle Linux 5 provides only OSS sound support. For a workaround to this problem, see [Knowledge Article 1464502.1](#).

13.1.7.7. Problem: An application ignores the \$AUDIODEV environment variable

Some applications fail to honor `$AUDIODEV` and unconditionally use a specific audio device node such as `/dev/audio` or `/dev/dsp`. To work around this shortcoming, Sun Ray Software provides a preloadable shared library `libc_ut.so` that can be used to interpose on an application and redirect its activities to the device specified by `$AUDIODEV`. To put this redirection into effect:

1. Navigate to the shell or wrapper from which you started the audio player.
2. Set the environment variable `LD_PRELOAD` in the player application's environment to refer to the `libc_ut.so` interposer:

```
$ LD_PRELOAD=libc_ut.so  
$ export LD_PRELOAD
```

3. Restart the application.

13.1.7.8. `xmms` Player Configuration (Oracle Linux)

To configure an `xmms` player to play mp3 files, perform the following steps:

1. Change the preferences on `xmms` output plugin to add more buffering.
2. Change the buffer size to 10000 ms and the Pre-Buffer percent to 90.

When you run `xmms`, from command line or menu, click on the `O` (letter O) on the left side of the panel to bring up the **Preferences** menu.

3. Under the **Audio I/O Plugins** button, select **Output Plugin OSS Driver** and click **Configure**.
4. Select Buffering.
 - a. The default Buffer size is 3000 ms. Change this to 10000 ms.
 - b. The default Pre-buffer percent is 25. Change this to 90.
5. Click **OK**, then click **OK** on the **Preferences** panel.
6. Exit `xmms` and restart it.

13.1.8. Audio Output Troubleshooting (Oracle Solaris 11 and Oracle Linux 6)

This section provides troubleshooting information for audio output on Sun Ray Clients and Oracle Virtual Desktop Clients running Oracle Solaris 11 or Oracle Linux 6 sessions.

The PulseAudio sound server is used to provide audio output for desktop client sessions when using these platforms. An instance of `utaudio`, running in PulseAudio mode, is created for each session to deliver audio from the PulseAudio server.

In this environment, applications do not use `$AUDIODEV` to send audio to the PulseAudio sound server. `$AUDIODEV` is applicable only if you need to open an audio device file, such as `/dev/audio` or `/dev/dsp`.

13.1.8.1. Additional Notes

Here are some additional notes and restrictions for audio output on Oracle Solaris 11 and Oracle Linux 6.

- To optimize the shared memory used by PulseAudio, add the following line to the `/etc/pulse/client.conf` file on the Sun Ray server:

```
shm-size-bytes = 131072
```
- When using Oracle Solaris 11, the Oracle Solaris 10 SADA audio interface is automatically available if needed for compatibility reasons. An additional SADA-specific `utaudio` process is started for each session. See [Section 13.1.7, “Audio Output Troubleshooting \(Oracle Solaris 10 and Oracle Linux 5\)”](#) for more details.
- For Oracle Solaris 11, the Adobe Flash Player is a SADA application that uses the `$AUDIODEV` interface. A device lockout may occur if multiple SADA applications share the same `$AUDIODEV`. To fix this problem, launch a separate `utaudio` command for each SADA application.
- For Oracle Solaris 11 Trusted Extensions, two audio devices should be displayed in the Device Allocation Manager, one for the SADA interface device and another one for PulseAudio. Make sure to allocate both audio devices.

13.1.8.2. Problem: Audio Output is not working (PulseAudio)

- Use the Oracle keyboard audio keys and check the volume and mute buttons.
- Display the Sun Ray session's audio settings:

```
$ utsettings
```

and verify that the audio output is selected properly, for example, for headphones or speakers.

- Make sure the volume is not muted in your desktop session.
- Try a set of external speakers plugged into the Sun Ray Client's audio out or headphones port. If that works, the Sun Ray Client might have a broken speaker.
- Make sure the `pulseaudio` process is running. If it was not started with the user session, remove the `$HOME/.pulse` directory, exit the session, and start a session again.
- Make sure there is a `utaudio` process running with the `-p` option to accept streaming from the PulseAudio sound server. If the process does not exist, check the `/var/opt/SUNWut/log/messages` log file to see if there is any error message from the `pulseaudio` or `utaudio` command.
- If audio does not work when using GStreamer-based programs, such as Totem or Rhythmbox, make sure GStreamer is configured to use PulseAudio by running `gststreamer-properties`. The PulseAudio Sound Server setting must be selected for both Audio Input and Output.
- For Oracle Linux 6, make sure ALSA is configured to use PulseAudio. To verify the configuration, make sure the `/etc/asound.conf` (server preference) or `$HOME/.asoundrc` file (user preference) includes the following:

```
pcm.pulse {
    type pulse
}
ctl.pulse {
```



```

    type pulse
}
pcm.!default {
    type pulse
}
ctl.!default {
    type pulse
}

```

If both files exist, the `$HOME/.asoundrc` file takes precedence.

- If audio does not work with OSS applications on Oracle Linux 6, make sure the `$AUDIODEV` environment variable is set to an instance of `utaudio`. For Oracle Linux 6, the `$AUDIODEV` environment variable is not defined by default. Here is an example of setting `$AUDIODEV`:

```
$ export AUDIODEV="/opt/SUNWut/bin/utaudio"
```

- OSS applications on Oracle Linux 6 can also work with PulseAudio by using the OSS wrapper (`padsp`):

```
$ padsp OSS_program
```

13.2. Sun Ray Clients

This chapter provides information about managing Sun Ray Clients and provides a list of typical procedures when using a Sun Ray Client. To start a session on a Sun Ray Client once it is properly configured, all you need to do is power on the Sun Ray Client, log in after it boots, and wait for the desktop to display.

The following chapters also provide more specific administration information for Sun Ray Clients:

- [Chapter 14, Sun Ray Client Firmware](#)
- [Chapter 16, Troubleshooting Icons](#)
- [Chapter 15, Peripherals](#)

13.2.1. How to Centralize Sun Ray Client Configurations (.parms)

Once a Sun Ray Client discovers the firmware server, it downloads its corresponding `*.parms` file. This file contains the firmware revision, which is checked against the client to make sure its firmware is up-to-date. The `.parms` file can also be used to centralize and provide other Sun Ray Client configuration values, such as the `servers` keyword used to specify the Sun Ray servers if the `sunray-servers` DNS entry isn't used. See [Table 13.4, "Sun Ray Client Configuration Parameters \(.parms\)"](#) for the list of configuration values.

By default, `utsetup` creates an initial set of `.parms` files for each Sun Ray model type. You can use the `-i` option of the `utfwadm` command and a template file to centralized updates for all the `.parms` file. The key/value entries in the template will be appended to each model-specific `.parms` file.

Steps

1. Become superuser on the Sun Ray server.
2. Change directory to the TFTP home directory.

This example uses the `/tftpboot` directory as the TFTP home directory.

```
# mkdir /tftpboot
```

3. Create the template for the `.parms` file.

This template is a text file with key/value pairs, and it can be located anywhere on your file system. It is common practice to store it in the TFTP home directory. In this example, the file is named `srsconfig`, and it resides in the `/tftpboot` directory.

```
# vi /tftpboot/srconfig
```

See [Table 13.4, “Sun Ray Client Configuration Parameters \(.parms\)”](#) for the list of key/value pairs that you can add to the `.parms` file.

4. Use the `utfwadm` command to update the `.parms` files.

The `utfwadm` command automatically uses the latest firmware installed on the Sun Ray server. Again, the following example uses `/tftpboot/srsconfig` for the template file location.

```
# /opt/SUNWut/sbin/utfwadm -AaV -i /tftpboot/srconfig
```



Note

You can also update the `.parms` file for a specific Sun Ray Client using the `-e MAC_address` option.

[Table 13.4, “Sun Ray Client Configuration Parameters \(.parms\)”](#) lists the key/value pairs.

Additional key/value pairs included in the `.parms` files are in `key=value` format, with case sensitivity and no spaces allowed. Options that take values of 0 or 1 have a default value of 0 if not specified. The following table lists the options that are allowed. For details on the options that can be used to configure the `.parms` files, see the `utfwadm` man page.

Table 13.4. Sun Ray Client Configuration Parameters (.parms)

Key	Description
<code>bandwidth=bits_per_second</code>	Sets the maximum bandwidth limit used by the Sun Ray Client, in bits per second.
<code>cmdcashsize=size</code>	Sets the command cache used to store the list of recent commands, in Kbytes. Default value is 512 Kbytes, maximum value is 8192 Kbytes, and a zero value disables the command cache.
<code>compress={0 1}</code>	When set to 1, forces compression on. Default is 1 (compression on).
<code>fastload={0 1}</code>	When set to 1, forces the maximum TFTP transfer size if the TFTP server supports it. The default is 512-byte packets. Over a high latency connection, using this setting typically doubles the speed of firmware downloads.
<code>fulldup={0 1}</code>	When set to 1, forces full duplex setting.
<code>enablegui={force none hashed-passwd prompt off}</code>	Enables or disables the Configuration GUI. These keywords should be changed only with the <code>utfwadm -G</code> or <code>utfwadm -g</code> command, respectively. See Chapter 14, Sun Ray Client Firmware for details.
<code>disablegui={force none hashed-passwd prompt off}</code>	
<code>kbcountry=code</code>	Forces the keyboard country code number (keyboard map) for a non-U.S. USB keyboard that reports a country code value of 0. This value can also be set on the Advanced menu of the Configuration GUI. See Section 13.2.10, “Keyboard Country Codes” for the list of code numbers.
<code>LogXXX=detail-level</code>	Sets the logging level for various classes of logging events, where <code>XXX</code> is one of <code>Appl</code> , <code>Vid</code> , <code>USB</code> , <code>Net</code> , or <code>Kern</code> . Valid values are 1 through 7, with 7 providing the most detailed logging output. A separate entry must be

Key	Description
	<p>specified for each type. The logging information is saved in the <code>/var/opt/SUNWut/log/messages</code> file on the LogHost.</p> <p>When using a shared network (LAN) with external DHCP server support (configured network using <code>utadm -L on</code>), logging for each event type is disabled unless the LogXXX value and the LogHost value are set.</p>
LogHost=	The Sun Ray server where logging output is saved when one or more LogXXX entries are specified. Valid value is a resolvable DSN host name or an IP address of a Sun Ray server. When using a failover group, the primary Sun Ray server should be specified.
lossless={0 1}	When set to 1, does not permit lossy compression to be used.
MTU=	Sets the network MTU. The value used is the minimum of those supplied from various sources.
orient1={0 90 180 270} orient2={0 90 180 270}	<p>Specifies the clockwise physical rotation, in degrees, of the monitors attached to the Sun Ray Client. <code>orient1</code> is for the first monitor and <code>orient2</code> is for the second monitor. If the key is not specified, the default is 0.</p> <p>This setting is used to provide the screen orientation to <code>Xnewt</code>, which in turn provides the value to the <code>utscreenresize</code> and <code>xrandr</code> commands to set a session's screen orientation accordingly.</p> <p>Note that video acceleration with Windows sessions is not currently supported with non-default screen orientations and should be disabled. You can use the <code>-B off</code>, <code>-F off</code>, and <code>-M off</code> options of the <code>uttsc</code> command to disable video acceleration.</p>
poweroff=	<p>Sets how much time a Sun Ray 3 Series Client will be idle before it turns off. The default power off time is 30 minutes. Setting <code>poweroff=0</code> disables the power off feature. When the power off feature is enabled, the firmware enforces a minimum power off value of 10 minutes and a maximum power off value of 30 days. The value for the power off feature is in minutes.</p> <p>For example, <code>poweroff=15</code> sets the idle power off timer to 15 minutes.</p>
select={inorder random}	Permissible values are <code>inorder</code> or <code>random</code> . Selects a server from the server list either starting at the beginning or at random, respectively.
servers=	Specifies a comma-separated mixture of host names or IP addresses indicating the available session servers.
stopkeys={keyn[-keyn]* none}	<p>Specifies an alternative combination of modifier keys to perform the same function as the Stop key on an Oracle keyboard or the Ctrl-Pause key sequence. By default, this alternative combination is Ctrl-Shift-Alt-Meta. See Section 13.2.2, "Sun Ray Client Hot Keys" for details.</p> <p>The value of <code>keyn</code> can be any combination of the Ctrl-Shift-Alt-Meta keys, but at least two of the keys must be used. For example, you can set this value to Ctrl-Alt or Meta-Ctrl-Shift.</p> <p>If this parameter is set to <code>none</code>, the alternative key combination is disabled.</p> <p>Note that the Meta key has different names on different keyboards: on a PC keyboard, it is the "Windows" key, and on a Mac keyboard, it is the "Command" key.</p>

Key	Description
stopqon={0 1}	When set to 1, enables the Stop-Q key sequence to be used to disconnect a Sun Ray Client from a server, in particular, if it's using a VPN connection.
utloadoff={0 1}	When set to 1, disables the ability to use the utload program to force a Sun Ray Client to load firmware.
videoindisable={0 1}	When set to 1, disables the input source on the front of a Sun Ray 270 Client, and it locks the monitor into displaying only the client output.

13.2.2. Sun Ray Client Hot Keys

For a Sun Ray Client, there are a number of predefined hot keys that can trigger an activity or event on a client, which are shown in [Table 13.5, "Sun Ray Client Hot Keys"](#). The key sequences can be either an Oracle-specific key combination (using keys that might exist only on Oracle keyboards) or by an alternative key combination that does not require Oracle-specific keys.

The activities controlled by these hot keys are specific to a Sun Ray Client. Desktop software running in the Sun Ray session might provide a separate keyboard shortcut facility that provides additional hot keys for desktop activities, perhaps including the ability to launch certain programs.

By default, the alternative prefix key combinations are Ctrl-Pause and Ctrl-Shift-Alt-Meta. The Ctrl-Pause key sequence is used throughout this documentation. The Ctrl-Shift-Alt-Meta key combination can be changed through the Advanced menu of the Configuration GUI (Enter Alternative STOP modifiers) or the `stopkeys` keyword in the `.parms` file. It can be set to any combination of the four keys, but at least two must be used.



Note

The Meta key has different names on different keyboards: on a PC keyboard, it is the "Windows" key, and on a Mac keyboard, it is the "Command" key.

Table 13.5. Sun Ray Client Hot Keys

Oracle-specific Hot Key	Non-Oracle Hot Key	Action
Mute	Ctrl-Pause-CursorDown	Mute and unmute audio.
Softer	Ctrl-Pause-CursorLeft	Decreases the audio volume.
Louder	Ctrl-Pause-CursorRight	Increases the audio volume.
Mute-Softer-Louder or Stop-N	Ctrl-Pause-N	Displays the Sun Ray Client's MAC and IP addresses and server IP address.
Ctrl-Power or Stop-A	Ctrl-Pause-A	Power cycles the Sun Ray Client. On an Oracle keyboard, the Power key has a crescent moon glyph and is positioned at the top right corner of the keyboard.
Stop-C	Ctrl-Pause-C	Clears any local configuration data on the Sun Ray Client.
Stop-O	Ctrl-Pause-O	Enables or disables the On-Screen Display (OSD) troubleshooting icons when a Sun Ray Client boots.
Stop-S or Stop-M	Ctrl-Pause-S or Ctrl-Pause-M	Opens the Configuration GUI to modify how to initialize the client. The Configuration GUI must be enabled on the client.
Stop-V	Ctrl-Pause-V	Displays the Sun Ray Client's model, MAC address, and firmware version.

Oracle-specific Hot Key	Non-Oracle Hot Key	Action
Ctrl-Alt-Bksp-Bksp	Ctrl-Alt-Bksp-Bksp	Terminates a session. This hot key cannot be reconfigured to another value, but it can be disabled. For details, see the utxconfig man page.
Ctrl-Alt-Del-Del	Ctrl-Alt-Del-Del	Terminates the process that has taken control of the X server.

There are also hot keys used to launch the [utsettings](#) or [utdetach](#) Sun Ray utilities. You can configure these hot key sequences through your `$HOME/.utslaunch.properties` file, or they can be set by the administrator per a site-wide basis. See [Section 13.2.2.1, “How to Configure the Utility Hot Keys”](#) for details.

13.2.2.1. How to Configure the Utility Hot Keys

Hot keys can be configured to launch the [utsettings](#) or [utdetach](#) Sun Ray utilities. The scopes for these hot keys are as follows:

- System-wide default setting
- User default setting
- System-wide mandatory setting

To support these levels of customization, the Sun Ray Client at session startup examines the following property files in the order shown in [Table 13.6, “Sun Ray Settings Property Files”](#).

Table 13.6. Sun Ray Settings Property Files

File	Scope	Description
/etc/opt/SUNWut/utslaunch_defaults.properties	System	This file contains the default properties. Any properties specified override any defaults built into the application itself.
\$HOME/.utslaunch.properties	User	This file contains the user's preferred values, which override any application or system-wide defaults.
/etc/opt/SUNWut/utslaunch_mandatory.properties	System	This file contains system-wide mandatory settings that cannot be overridden by the user. These properties override any application, system-wide, or user defaults.

If your policy is for all users to use the same standard hot key, modify the system-wide mandatory defaults file to specify this standard key. This setting prevents users from specifying their own hot key preferences.

The format of the hot key entry in these property files is `utility_name.hotkey=value`, where `utility_name` is the name of the utility (currently either [utsettings](#) or [utdetach](#)) and `value` is a valid X keysym name preceded by one or more of the supported modifiers ([Ctrl](#), [Shift](#), [Alt](#), [Meta](#)) in any order. Default values are shown in [Table 13.7, “Defaults for Configurable Hot Key Values”](#).

Table 13.7. Defaults for Configurable Hot Key Values

Configuration Property Name	Default Hot Key	Action
utsettings.hotkey	Shift-Props	Invokes the Sun Ray Settings GUI.
utdetach.hotkey	Shift-Pause	Detaches the session from this Sun Ray Client. (Often used to detach a non-smart card mobility session.)

13.2.2.1.1. How to Change Utility Hot Key Settings for All Users

If you don't want your users to use the default hot keys to launch the utilities, you can set up the system-wide defaults file to specify different hot keys. Users can still specify their preferences in the user defaults file.

1. As superuser, open the `/etc/opt/SUNWut/utslaunch_defaults.properties` file in a text editor.

**Note**

If you want to make the change mandatory for all users even if they have user defaults set, change the value in the `/etc/opt/SUNWut/utslaunch_mandatory.properties` file.

2. Locate the original hot key entry for the utility you want to change and place a `#` in front of it to comment it out.

For example:

```
# utdetach.hotkey=Shift Pause
```

3. Type the new hot key property after the first statement.

For example:

```
utdetach.hotkey=Alt F9
```

4. Save the `utslaunch_defaults.properties` file.
5. Log out and log back in to enable the new hot key.

13.2.2.1.2. How to Change the Utility Hot Key Settings for a Single User

A user's hot key settings override any system-wide default settings, unless they are mandatory.

1. In the user's home directory, create the `.utslaunch.properties` file.

**Note**

Make sure that the user owns and can read this file.

2. Add a line to the `.utslaunch.properties` file with the value for the hot key.

For example:

```
utsettings.hotkey=Shift F8
```

3. Save the `.utslaunch.properties` file.
4. Log out and log back in to enable the new hot key.

13.2.3. How to Change Sun Ray Client Audio and Display Settings (Sun Ray Settings GUI)

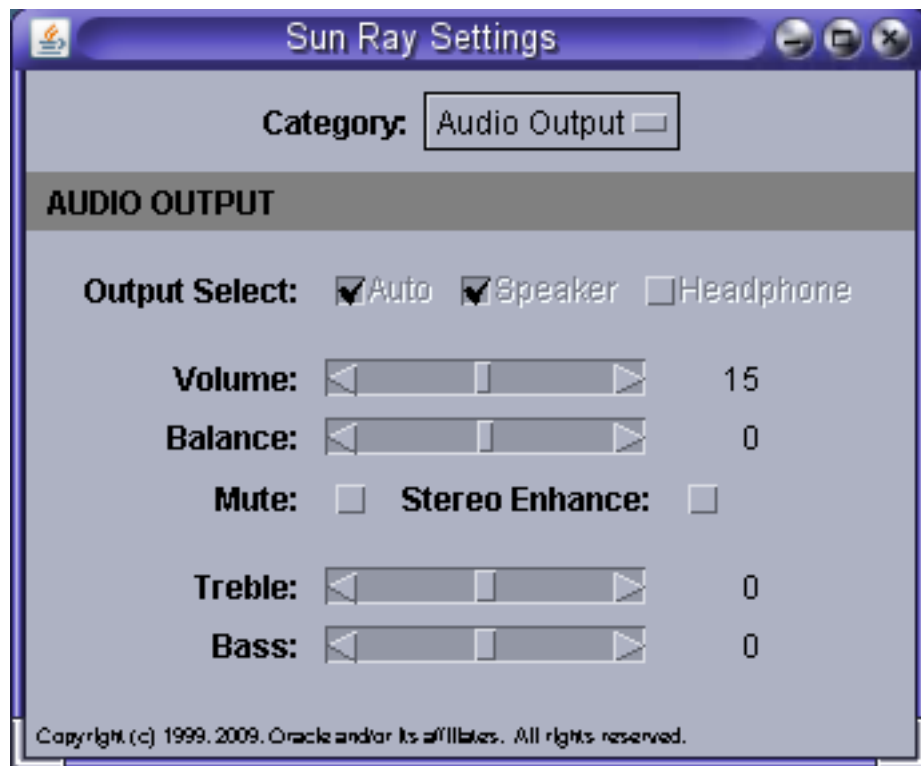
The Sun Ray Settings GUI enables you to view and change the audio and display settings for a Sun Ray Client that you are currently logged into. The `utset` command provides a non-GUI mechanism for reporting and modifying Sun Ray Client settings. For details, refer to the `utset` man page.

1. Press the Settings hot key or run the `utsettings` command.

The default Settings hot key combination is Shift-Props, but this assignment can be reconfigured as described in [Section 13.2.2.1, “How to Configure the Utility Hot Keys”](#).

The Sun Ray Settings GUI is displayed, as shown in [Figure 13.1, “Sun Ray Settings GUI”](#).

Figure 13.1. Sun Ray Settings GUI



2. Use the **Category** menu to view the **Audio Output**, **Audio Input**, **Display**, or **Video** settings panels.
3. To change a setting, move the appropriate scroll bar, checkbox, or pull-down menu.

Changes to the monitor signal timing through the **Resolution/Refresh Rate** setting require confirmation before and after the change is applied to the client. All other changes take effect immediately.

4. Dismiss the Sun Ray Settings GUI.
 - If the window was launched by the Settings hot key, press the hot key again or apply the window manager's **close** action to that window.
 - If the window was launched by invoking `utsettings` directly, apply the window manager's **close** action to that window.

13.2.4. How to Modify Screen Resolutions

You can modify a Sun Ray Client's screen resolution settings by invoking the `utsettings` command.

Any resolution selection made within a session remains effective whenever the session is displayed on that particular Sun Ray Client. The selection is not lost if the unit goes into power-save mode or is power-

cycled; however, the resolution settings selected through the `utsettings` command apply only to the client where the command is run.

When you move to another Sun Ray Client, the resolution settings do not accompany you to the new client, but the settings remain effective for your session on the original client if you return to the session through hotdesking.

If the session is associated with a personal mobile token, such as a smart card or an NSCM credential, a message displays offering to make the selected timing permanent. If you accept that offer, then the timing is retained and reused on your subsequent personal mobile token sessions on the same client.

In addition, an administrator can use the `utresadm` command to arrange for particular monitor timing to be used in the following situations:

- Whenever a specific token is presented on a specific client
- On a specific client regardless of the token that is presented at the client
- On all clients regardless of the token that is presented at the client

Any conflict among settings is resolved in favor of the most specific configuration rule. That is, a configuration record for a specific token at a specific Sun Ray Client takes precedence over a record for any token at that specific client, and a configuration record for any token at a specific client takes precedence over a record for any token at any client.

For further details, see the `utsettings` and `utresadm` man pages.

13.2.5. How to Power Cycle a Sun Ray Client

To power cycle a Sun Ray Client with a hard reset:

- Disconnect and then reconnect the power cord.
- Press the power button if one is available.

To power cycle a Sun Ray Client with a soft reset, press Ctrl-Power or Stop-A.

The Power key at the right side of the top row of an Oracle Type 6 or Type 7 keyboard has a crescent moon icon. Therefore, the soft reset key sequence is often called Ctrl-Moon.

13.2.6. How to Enable or Disable XRender

Sun Ray Software includes the Xserver process, Xnewt, as the default Xserver. Xnewt also includes the capability to use the X Rendering Extension (Render), which allows applications on a client to use a rendering model based on Porter-Duff compositing. XRender is enabled by default because many new X applications require XRender to improve performance or to even function properly.

However, some applications use of XRender may conflict with optimizations in the Sun Ray protocol and create an increase in both CPU loading and network bandwidth consumption. In these instances, the applications may see a performance benefit by disabling the XRender extension.

By default, XRender is enabled. If a Sun Ray Client experiences performance degradation with a particular application, use the following procedure to disable XRender.



Note

After enabling or disabling XRender, you must restart your current Sun Ray session (Ctrl-Alt-Bksp-Bksp) for the change to take affect. Or, you can log out from your current session and log back in.

To disable XRender on a client:

```
% utxconfig -n off
```

To enable XRender on a client:

```
% utxconfig -n on
```



Note

You can use the `-A` option to supersede all user configured and system default settings.

13.2.7. How to Configure Screen Rotation

You can configure the orientation of the monitors connected to a Sun Ray Client, so a session's screen automatically rotates to the appropriate monitor orientation. For example, if the connected monitors are in portrait mode, setting the orientation for these monitors will ensure that the session's screen will automatically output in portrait mode. The default orientation is the standard landscape position.



Note

The On-Screen Display (OSD) icons will not automatically rotate if a non-default orientation is configured. Also, video acceleration with Windows sessions is not currently supported with non-default screen orientations and should be disabled. You can use the `-B off`, `-F off`, and `-M off` options of the `uttsc` command to disable video acceleration.

See [Section 13.2.1, “How to Centralize Sun Ray Client Configurations \(.parms\)”](#) for details on using the `orient1` and `orient2` keywords to configure monitor orientation.

13.2.8. How to Disable Screen Blanking on a Sun Ray Client

There may be times when you do not want your Sun Ray Client in power saving mode, during which the screen goes blank after a specific period of non use. Power management is a feature of the Sun Ray Software and it is enabled by default.

You may also want to disable the desktop's power management or screensaver feature.

To Disable Screen Blanking at the Sun Ray Client Level

Set the **Advanced > Video > Blanking** parameter to `0` in the Configuration GUI, if enabled. For more details, see [Section 14.5, “How to Modify a Sun Ray Client's Local Configuration \(Configuration GUI\)”](#).

To Disable Screen Blanking at the Desktop Level

Refer to your desktop documentation about how to disable the power management feature or screensaver feature.

Here are some examples:

- Use the `xset s noblank;xset s 0 0;xset -dpms;xset s off` command.
- For Oracle Solaris, make sure that `xscreensaver` is disabled or configured to not blank or lock the screen. If active, `xscreensaver` overrides any settings you have made using the `xset` command. See the `xscreensaver(1)` man page for details.

Here is a sample `.xscreensaver` configuration file that you can add to the root of your home directory to disable `xscreensaver`.

```
# Parameters to disable Xscreensaver
timeout: 0
cycle: 0
lock: False
dpmsEnabled: False
mode: off
```

- For Oracle Linux, make sure that [gnome-screensaver](#) is disabled or configured to not blank or lock the screen. You can use the [gnome-screensaver-command --exit](#) command to disable the screensaver for the current session or add it to a session script. See the [gnome-screensaver-command\(1\)](#) man page for details.

You can also use the following script to permanently disable [gnome-screensaver](#) for a user through the [GConf](#) settings.

```
#!/bin/sh
# Script to disable gnome-screensaver, including keybindings,
# and power management UI icon

theGConf="/usr/bin/gconftool-2";export theGConf
BOOL="--type bool --set";export BOOL
STRING="--type string --set";export STRING
INT="--type int --set";export INT
UNSET="--recursive-unset";export UNSET

$theGConf $UNSET /apps/gnome_settings_daemon/keybindings/screensaver
$theGConf $BOOL /apps/gnome-screensaver/embedded_keyboard_enabled false
$theGConf $STRING /apps/gnome-screensaver/mode "blank-only"
$theGConf $BOOL /apps/gnome-screensaver/lock_enabled false
$theGConf $BOOL /apps/gnome-screensaver/idle_activation_enabled false
$theGConf $INT /apps/gnome-screensaver/lock_delay "0"
$theGConf $STRING /apps/gnome-power-manager/ui/icon_policy "never"
$theGConf $INT /apps/gnome-power-manager/ac_sleep_display "0"
```

13.2.8.1. How to Disable Screen Blanking for All Sun Ray Sessions

The following procedure describes how to disable screen blanking for all Sun Ray sessions.



Note

This configuration works for all kiosk sessions, but it does not work for sessions using the GNOME Display Manager through a regular session mode.

1. Become superuser on the Sun Ray server.
2. Change directory to the session initialization directory.

- Oracle Solaris 10:

```
# cd /usr/dt/config/Xsession.d
```

- Oracle Linux or Oracle Solaris 11:

```
# cd /etc/X11/xinit/xinitrc.d
```

3. Create the following customized script (the script is called [0050.utblank.sh](#) in this procedure).

```
#!/bin/sh

# This script disables the X Server from blanking
# for both Oracle Solaris and Oracle Linux
```

```
# Check for OS and add the path to xset for Oracle Solaris

TheOS=`uname`
if [ "$TheOS" = "SunOS" ];then
PATH=$PATH:/usr/openwin/bin:export PATH
fi

# Disable the Xserver from screen blanking and
# disable Display Power Management Signaling

xset s 0 0
xset s noblank
xset -dpms
xset s off

# End of Script
```

**Note**

The script name should have the `0050.` prefix to make sure it is run at the appropriate time.

**Note**

For Oracle Linux and Oracle Solaris 11, the script name must have the `.sh` extension, otherwise the script will not get sourced.

4. Save the script and make the script executable for everyone.

```
# chmod 775 0050.utblank.sh
```

5. Start a new session, so the script gets sourced.

13.2.9. How to Enable the NumLock Key for All Sun Ray Sessions

The `utkeylock` command can modify the state of certain locking modifier keys on a user's keyboard. Currently, only the NumLock key is supported. This command may be useful to invoke during session creation to enable NumLock for users who expect NumLock to be on by default, which is typical for Windows PCs. By default, the NumLock key is disabled on a Sun Ray Client.

**Note**

This configuration works for all kiosk sessions, but it does not work for sessions using the GNOME Display Manager through a regular session mode.

The following procedure describes how to enable the NumLock key for all Sun Ray sessions.

1. Become superuser on the Sun Ray server.
2. Change directory to the session initialization directory.

- Oracle Solaris 10:

```
# cd /usr/dt/config/Xsession.d
```

- Oracle Linux or Oracle Solaris 11:

```
# cd /etc/X11/xinit/xinitrc.d
```

3. Create one of the following customized scripts based on the result you want (the script is called `0050.utnumlock.sh` in this procedure).

- Enable the NumLock key when a session initializes.

```
#!/bin/sh

# Enable NumLock key for each session
/opt/SUNWut/bin/utkeylock -n on
```

- Enable the NumLock key when a session initializes and on all subsequent connections through hotdesking.

```
#!/bin/sh

# Enable NumLock key on and make sure it stays on each time a user hotdesks
/opt/SUNWut/bin/utaction -i -c "/opt/SUNWut/bin/utkeylock -n on" &
```



Note

The script name should have the `0050.` prefix to make sure it is run at the appropriate time.



Note

For Oracle Linux and Oracle Solaris 11, the script name must have the `.sh` extension, otherwise the script will not get sourced.

4. Save the script and make the script executable for everyone.

```
# chmod 775 0050.utnumlock.sh
```

5. Start a new session, so the script gets sourced.

13.2.10. Keyboard Country Codes

A keyboard country code is a number representing a specific USB keyboard map that can be set in the Sun Ray Client firmware to provide better non-US keyboard support. This code is needed if the keyboard returns a country code of 0.

The code can be set through the Configuration GUI or the `.parms` file.



Note

There is currently no way to get the keyboard type information for a Sun Ray Client.

This is the list of valid keyboard country codes.

- 1 Arabic
- 2 Belgian
- 3 Canada_Bi
- 4 French-Canadian
- 5 Czech
- 6 Denmark
- 7 Finnish

- 8 France
- 9 Germany
- 10 Greek
- 12 Hungarian
- 14 Italy
- 15 Japan
- 16 Korea
- 17 Latin-American
- 18 Netherland
- 19 Norway
- 21 Polish
- 22 Portugal
- 23 Russia
- 24 Slovakian
- 25 Spain
- 26 Sweden
- 27 Switzerland
- 28 Switzerland_Ge
- 30 Taiwan
- 31 TurkeyQ
- 32 UK-English
- 33 US-English
- 35 TurkeyF

13.2.11. Sun Ray Client Boot Process

This process flow shows how a Sun Ray Client obtains its basic network parameters, firmware server, and session server. Many of the configuration options listed in this process flow are described in [Chapter 19, Alternate Network Configurations](#).

**Note**

The [Configuration GUI](#) must be enabled on the client for the user to locally configure the Sun Ray parameters. Locally configured parameter values override network values with the exception of MTU, which is always the minimum of the values seen.

1. Power unit on.

2. Read local configuration (from Configuration GUI), if present.
 - a. netType = STATIC IP **OR** DHCP **OR** Auto-config (IPv6)
 - b. If netType is STATIC IP, use locally configured values for
 - IP Address
 - Net mask
 - Broadcast address
 - Router
 - MTU
3. Bring up the network interface.
 - a. If any networking values missing, then perform DHCP.
 - b. If AuthSrvr value is not defined, then perform DHCP_INFORM request.
 - c. Merge any local values, DHCP vendor options, and DHCP_INFORM values (local values override DHCP except MTU, which is minimum of values seen).
 - d. If XDispMgr was given by DHCP **AND** no AltAuth vendor option was found, then set AltAuth to XDispMgr (option 49) values.
4. Read Configuration Parameter file (*model.parms* file) on firmware server.
 - a. Try to find the firmware servers that contain *.parms* file, in order:
 - i. Locally configured value
 - ii. DHCP vendor option (FWSrvr)
 - iii. Option 66 (TftpSrvr) IP Address or DNS name
 - iv. DNS lookup of "sunray-config-servers" (if mapped to multiple addresses, choose one randomly)
 - b. Download the *.parms* file.
 - i. Search for SunRayPx.MAC.parms.
 - ii. Search for SunRayPx.parms.
 - c. Parse the *.parms* file.
 - parms.version = firmware version
 - parms.revision = max supported hardware revision
 - parms.barrier = barrier value of server firmware
 - parms.BarrierLevel = barrier override value
 - parms.servers = server list
 - parms.select = inorder | random

- d. If `.parms` file was successfully parsed **OR** firmware server was obtained by *locally configured value*, then go to Step 5.



Note

If a locally configured firmware server is unreachable or the correct configuration parameter file does not exist, the Sun Ray Client will not attempt any of the other methods in Step 4 to locate configuration parameter files. This setup prevents the unintentional loading of a different firmware version than is provided by the locally designated firmware server.

- e. If no `.parms` file found **AND** not at end of firmware server list, then go to Step 4 and pick next firmware server on the list.
- f. If no firmware servers left to try, then set following values:
- `parms.version` = DHCP vendor option NewTVer (set to NULL string if none provided by DHCP)
 - `parms.BarrierLevel` = DHCP BarrierLevel (set to `current_barrier` if none provided by DHCP)
 - set `parms.revision` to `current_revision`
 - set `parms.barrier` to `current_barrier`
 - set `parms.select` = `inorder`
5. Determine if there is new firmware to load.
- If:
- `parms.version` is not equal to the current firmware version
 - **AND** `parms.version` is not equal to `"_NONE_"`
 - **AND** `parms.revision` is `>=` to current hardware revision
 - **AND** either `parms.barrier` is `>=` to `parms.BarrierLevel` or `parms.barrier` is `>=` current firmware's barrier level
- Then:
- a. Download firmware.
 - b. Write firmware to flash.
 - c. Reboot.
- Else:
- No firmware is loaded.
6. Determine a Sun Ray server to connect to.
- a. If `AlthAuth/AuthSrvr/parms.servers` are all empty, then set `server_list` to `"sunray-servers"`. Otherwise set `server_list` to `parms.servers`.
 - b. If untried `server_list` addresses are left, then:

- i. Select a name in order (or randomly if parms.select=random).
 - ii. Translate the name to a list of IP addresses (either DNS lookup, or IP address notation).
 - iii. Select an address from the list in order (or randomly if parms.select=random).
 - iv. Set that the broadcast address was seen if the selected address is the broadcast address, and select the next address.
 - v. Go to Step 6h.
 - c. If untried AltAuth addresses are left, then:
 - i. Select an address in order (or randomly if parms.select=random).
 - ii. Set that the broadcast address was seen if the selected address is the broadcast address, and select the next address
 - iii. Go to Step 6h.
 - d. If AuthSrvr is defined, then:
 - i. Set address to AuthSrvr.
 - ii. Go to Step 6h.
 - e. If broadcast address was seen, then perform broadcast protocol.
 - f. If broadcast response received, then:
 - i. Set selected address to responder.
 - ii. Go to Step 6h.
 - g. Timeout in 30 seconds and reboot.
 - h. Try to connect to selected address.
 - i. If connection fails, then go to Step 6b.
7. Sun Ray Client is connected.

13.3. Oracle Virtual Desktop Clients

This chapter describes how to enable access for Oracle Virtual Desktop Clients, explains client ID differences with Sun Ray Clients, and provides troubleshooting information.

13.3.1. Oracle Virtual Desktop Clients Overview

An Oracle Virtual Desktop Client is a software version of a Sun Ray Client. The Oracle Virtual Desktop Client application runs on an ordinary PC or tablet and provides a Sun Ray session in a desktop window. It is supported and can be installed on Windows, Linux, Mac OS X, iPad, and Android. An Oracle Virtual Desktop Client supports most of the standard Sun Ray Client functionality.

Instead of relying only on a Sun Ray Client for session access, a user can install and run the Oracle Virtual Desktop Client application on a laptop or a desktop. This is useful, for example, if users use a Sun

Ray Client at the office, but they want to access the same Sun Ray session at home from their laptop or desktop.

**Note**

Most Sun Ray Client references in this document also apply to Oracle Virtual Desktop Clients, unless otherwise specified.

The Oracle Virtual Desktop Client product is not provided with the Sun Ray Software media image. It must be downloaded separately, which is available from the Sun Ray Products Download page at <http://www.oracle.com/technetwork/server-storage/sunrayproducts/downloads/index.html>.

See the Oracle Virtual Desktop Client documentation (<http://www.oracle.com/technetwork/server-storage/sunrayproducts/docs/index.html>) for detailed information about the Oracle Virtual Desktop Client application.

13.3.2. Using External Devices on the Client Computer

You can access many of the external devices connected to the client computer when using Oracle Virtual Desktop Clients. In most cases, device access is automatic and can be configured. For Oracle Virtual Desktop Client 3.x or later, you can configure device access through the **Settings** tab. Here is the list of devices that you can access through Oracle Virtual Desktop Client, including any configuration steps required on the Sun Ray server.

- **Keyboard and pointing devices** - Provides access to the available keyboard and pointing device connected to the client computer. You can configure the keyboard through the **Settings** tab.
- **Audio devices** - Provides access to one input and one output audio device connected to the client computer. You can choose which audio devices to use through the **Settings** tab.
- **Smart card readers** - Provides access to a single smart card reader connected to the client computer. You can choose which smart card reader to use through the **Settings** tab.
- **Serial devices** - Provides access to serial devices connected to the client computer running the Windows operating system. Serial devices on Mac OS X and Linux computers cannot be accessed. Serial devices are automatically mounted and made available to Oracle Solaris and Oracle Linux sessions, and you must configure logical mappings to the serial devices when using the Windows connector. See [Section 15.4, “Accessing Serial Devices and USB Printers”](#) and [Section 17.17, “Accessing Serial Devices”](#) for details.
- **USB devices** - Provides access to any USB device that is attached to the client computer when using the Windows connector and Oracle Virtual Desktop Client 3.2 or later. You can choose which USB devices to use through the **Settings** tab. If you choose a USB device to use, that device is disconnected from the local environment, so it can be used by Oracle Virtual Desktop Client. The device is no longer available to the local system. As with Sun Ray Clients, USB devices that use isochronous interfaces will not work.

See the [Oracle Virtual Desktop Client User Guide](#) for more detailed information about accessing devices on the client computer.

13.3.3. How to Enable Access for Oracle Virtual Desktop Clients

This procedure describes how to enable access for Oracle Virtual Desktop Clients either through the `utpolicy` command or the Admin GUI. By default, access for Oracle Virtual Desktop Clients is disabled on a Sun Ray server. The on-screen display (OSD) 47 icon is displayed if users try to use Oracle Virtual

Desktop Client and access is currently disabled. See [Section 13.3.5, “Oracle Virtual Desktop Client Troubleshooting”](#) for more information.

You might also need to configure firewall settings as follows:

- **Client computers.** Ensure that firewall settings on the client computers allow Oracle Virtual Desktop Client to access the Internet.
- **Sun Ray servers.** See [Section 3.1.14, “Ports and Protocols”](#) for information on the ports used by Oracle Virtual Desktop Clients.

Oracle Virtual Desktop Clients can be used to access both smart card sessions and non-smart card sessions. Session mobility, or hotdesking, is supported with or without smart cards.

**Note**

The following procedures use a warm restart of Sun Ray services. If you *disable* access for Oracle Virtual Desktop Clients, use a *cold restart*.

Admin GUI Steps

1. Click the Advanced tab.
2. Click the Security subtab.
3. Select the **Oracle Virtual Desktop Client** option in the Card Users and Non-Card Users sections.
This enables both smart card and non-smart card sessions.
4. Click **Warm Restart** on the Servers page to restart Sun Ray services.

Command Line Steps

1. View the current policy.

Use the `utpolicy` command, as follows:

```
# /opt/SUNWut/sbin/utpolicy
Current Policy:
-a -g -z both -M
```

**Note**

The `-M` option enables non-smart card mobile (NSCM) sessions.

2. Edit the current policy, to enable access for Oracle Virtual Desktop Clients.

Do one of the following:

- a. To enable both smart card and non-smart card sessions, add the `-u both` option to your policy options.

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

- b. To enable only non-smart card sessions, add the `-u pseudo` option to your policy options.

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

- c. To enable only smart card sessions, add the `-u card` option to your policy options.

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

3. Restart the Sun Ray services.

```
# /opt/SUNWut/sbin/utstart
```

After enabling or disabling access for Oracle Virtual Desktop Clients, a restart of Sun Ray services in the server group is required.

13.3.4. How to Enable the Clipboard Service for Oracle Virtual Desktop Clients

This procedure describes how to enable copy and paste text between an application running in an Oracle Virtual Desktop Client session and an application running on the local desktop. When enabled, Oracle Virtual Desktop Client users can copy and paste text between an application running in an Oracle Virtual Desktop Client session and an application running on the local desktop. Copying and pasting Unicode characters is supported.

For the copy and paste functionality to work, the clipboard service must be enabled on the Sun Ray server and clipboard sharing must be enabled on Oracle Virtual Desktop Client running on the client computer. You can use the `utdevadm` command or the Advanced>Security page on the Admin GUI to check if the clipboard service is enabled.



Note

This feature is not available on Sun Ray servers running Oracle Solaris Trusted Extensions.

Follow these steps to enable the clipboard service on the Sun Ray server.

Admin GUI Steps

1. Click the Advanced tab.
2. Click the Security subtab.
3. Select the **Oracle Virtual Desktop Client Clipboard** option in the Devices section.
4. Restart Sun Ray services using the **Warm Restart** button on the Servers page.

Command Line Steps

1. Become superuser on the Sun Ray server.
2. Enable the clipboard service on Oracle Virtual Desktop Clients.

```
# /opt/SUNWut/sbin/utdevadm -e -s clipboard
```

3. Restart the Sun Ray services.

```
# /opt/SUNWut/sbin/utstart
```

13.3.5. Oracle Virtual Desktop Client Troubleshooting

This section provides troubleshooting information for Oracle Virtual Desktop Clients. Any troubleshooting information that involves the command line is not applicable to tablets running Oracle Virtual Desktop Client.

13.3.5.1. Connection Problems When Using a VPN or WAN

The Maximum Transmission Unit (MTU) is the maximum packet size for connections. By default, the MTU is set to 1500 bytes.

If you experience problems when using a Virtual Private Network (VPN) or a wide area network (WAN), the MTU setting might be too high for your network.

To diagnose the correct MTU setting for your network, use the `ping` command to find the largest packet size that can be transmitted successfully.

On Windows platforms:

```
ping server-name -l bytes -f
```

where `server-name` is the name of the Sun Ray server and `bytes` is the packet size.

On Mac OS X platforms:

```
ping -s bytes -D server-name
```

where `server-name` is the name of the Sun Ray server and `bytes` is the packet size.

On Linux platforms:

```
ping server-name -s bytes
```

where `server-name` is the name of the Sun Ray server and `bytes` is the packet size.

To calculate the MTU setting, add eight bytes to the packet size.

To set the MTU, either change the setting on the Network tab or run the following command:

```
ovdc --mtu bytes server-name
```

where `bytes` is the MTU, in bytes and `server-name` is the name of the Sun Ray server.

13.3.5.2. Screen Rendering Problems

Screen rendering problems, such as slow screen redraw or blocks of black pixels, can occur if the Maximum Transmission Unit (MTU) setting is too high for the network.

The MTU is the maximum packet size for connections. By default, the MTU is set to 1500 bytes.

See [Section 13.3.5.1, “Connection Problems When Using a VPN or WAN”](#) for details of how to diagnose the correct MTU setting for your network.

13.3.5.3. How to Set the Logging Level

To help you to diagnose problems with Oracle Virtual Desktop Clients, you can increase the logging level.

[Table 13.8, “Oracle Virtual Desktop Client Logging Levels”](#) shows the available logging levels.

Table 13.8. Oracle Virtual Desktop Client Logging Levels

Level	Description
0	No logging
1	Critical messages

Level	Description
2	Warnings
3	Informational messages

By default, the logging level is `0`, which sets logging to off. You can also set the logging domains (categories to log) with the `--logging-domains` option, but all logging domains are logged by default.

The logging level is cumulative. For example, the maximum logging level `3` includes informational messages, warnings, and critical messages.

To set the logging level, run the following command:

```
ovdc --logging-level num server-name
```

where `num` is the logging level and `server-name` is the name of the Sun Ray server.

For example, to record warnings and critical messages for a connection to the `sr-1.example.com` Sun Ray server, run the following command:

```
ovdc --logging-level 2 sr-1.example.com
```

13.3.5.4. How to Change the Log File Location

By default, log messages are written to a `.log` text file on the client computer. The `.log` file is named after the profile used. For example, the log file for the default profile is called `default.log`.

The default location of the log file depends on the installation platform, as follows:

- **Microsoft Windows XP platforms** – `C:\Documents and Settings\username\Application Data\OVDC\profilename.log`
- **Microsoft Windows 7 and Microsoft Windows 8 platforms** – `C:\Users\username\AppData\Roaming\OVDC\profilename.log`
- **Mac OS X platforms** – `$HOME/.OVDC/profilename.log`
- **Linux platforms** – `$HOME/.OVDC/profilename.log`

If you use the `--profile` command option to specify the path to a profile, the log file is created automatically in the same directory as the profile. In the following example, log messages are written to the `C:\temp\fullscreen.log` file.

```
ovdc --profile C:\temp\fullscreen
```

13.3.5.4.1. Using the `--logfile` Command Option

You can use the `--logfile` command option to change the name and location of the log file. If the path to the log file contains spaces, surround the path with straight quotation marks (").

The following example uses the default profile and writes log messages to the `mylog.txt` file in the default location.

```
ovdc --logfile mylog.txt
```

The following example uses the default profile and writes log messages to the `C:\temp\logfile.txt` file.

```
ovdc --logfile C:\temp\logfile.txt
```

The following example uses the `C:\profiles\fullscreen` profile and writes log messages to the `C:\temp\logfile.txt` file.

```
ovdc --profile C:\profiles\fullscreen --logfile C:\temp\logfile.txt
```

The following example uses the `C:\profiles\fullscreen` profile and writes log messages to the `mylog.txt` file in the default location.

```
ovdc --profile C:\profiles\fullscreen --logfile mylog.txt
```

13.3.5.5. How to Diagnose Connection Problems

Sun Ray Software uses a combination of the on-screen display (OSD) icons and localized error messages to display the status of a connection. The OSD icons can be used to diagnose connection problems with Oracle Virtual Desktop Clients.

The following OSD icon is displayed if users try to use Oracle Virtual Desktop Client and access is currently disabled.

Figure 13.2. Access Not Enabled OSD Icon



See [Section 13.3.3, “How to Enable Access for Oracle Virtual Desktop Clients”](#) to fix this problem.

See [Chapter 16, *Troubleshooting Icons*](#) for details about all the available OSD icons.

Chapter 14. Sun Ray Client Firmware

Table of Contents

14.1. Firmware Overview	177
14.2. Firmware Server Discovery	178
14.3. How to Update Firmware on Sun Ray Clients	178
14.4. How to Enable and Disable the Configuration GUI on All Sun Ray Clients	179
14.5. How to Modify a Sun Ray Client's Local Configuration (Configuration GUI)	181
14.5.1. Security Configuration Repository	181
14.5.2. Configuration GUI Menu Descriptions	182
14.5.3. How to Load a Remote Configuration File	187
14.6. VPN Support	189
14.6.1. How to Configure VPN Using Cisco Hybrid Authentication	190
14.7. IPsec	190
14.8. 802.1x Authentication	191
14.8.1. How to Configure and Enable 802.1x Authentication on a Sun Ray Client	191
14.9. How to Display Firmware Versions for All Currently Connected Sun Ray Clients	193
14.10. How to Display the Firmware Version from a Sun Ray Client	193
14.11. How to Synchronize the Sun Ray Client Firmware	193
14.12. How to Downgrade Firmware on a Sun Ray Client	193
14.13. How to Disable All Sun Ray Client Firmware Updates	195

This chapter describes how to manage the Sun Ray Operating Software (firmware) on Sun Ray Clients. The Sun Ray Operating Software must be downloaded and installed separately from My Oracle Support. The firmware is officially called Sun Ray Operating Software, but the term "firmware" will continue to be used throughout the documentation.

14.1. Firmware Overview

Every Sun Ray Client contains a firmware module that handles the following items:

- Power-on self test (POST)
- Client initialization
- Authentication
- Low-level input and output, such as keyboard, mouse, and display information.

To accommodate customers with differing requirements for flexibility and security, the Sun Ray Client firmware can be used in two ways:

- Default mode where the firmware uses the configuration provided by the Sun Ray server's `.parms` file.
- Local configuration mode where the firmware uses the Sun Ray Client's local configuration that a user can update through a Graphical User Interface (GUI) tool, called the Sun Ray Local Configuration GUI or Configuration GUI (previously called the Pop-up GUI).

A Sun Ray Client's local configuration values are checked first during the Sun Ray Client initialization, so this mode enables users to individually configure a Sun Ray Client's behavior at the local level. To make the Configuration GUI available to users on Sun Ray Clients, you must specifically enable it with the `utfwadm` command.

14.2. Firmware Server Discovery

Starting with Sun Ray Software 5.3, the firmware for Sun Ray Clients (called Sun Ray Operating Software) must be downloaded and installed separately on Sun Ray servers. Any Sun Ray server providing the latest Sun Ray Operating Software for Sun Ray Clients is considered a firmware server. For details, see [Section 3.2.4, “Installing Firmware Before Sun Ray Software Installation”](#).

When a Sun Ray Client boots in a properly configured environment, it checks with a firmware server to determine if it needs a Sun Ray Operating Software update. A Sun Ray Client's firmware server is discovered in the following order:

1. Locally configured value (configured through Configuration GUI)
2. DHCP Sun Ray vendor option (FWSrvr)
3. Generic DHCP option 66 (TftpSrvr) (IP Address or DNS name)
4. DNS lookup of `sunray-config-servers` (if mapped to multiple addresses, choose one randomly)

Each of these values are attempted in order until one succeeds. Although it is the last value attempted, the DNS lookup is the recommended firmware discovery configuration, as described below.

If the local configuration value is used and fails, none of the others are attempted. This prevents the overwriting of custom-configured firmware in a situation where the controlling firmware server happens to be temporarily unresponsive. See [Section 13.2.11, “Sun Ray Client Boot Process”](#) for more details on how a Sun Ray Client finds its firmware server.

Once a firmware server is discovered by a Sun Ray Client, the client retrieves a parameter file (`.parms`) via TFTP. This file is used by the client to determine if its currently installed Sun Ray Operating Software is older than the version on the firmware server. If so, the newer firmware is automatically downloaded and installed on the client.

In the event of an error in the firmware download, error messages through OSD display icons (if enabled) provide additional information that can be useful in diagnosing and correcting the problem. See [Chapter 16, *Troubleshooting Icons*](#) for details.



Note

By default, a client's firmware uses the configuration provided by the Sun Ray server's `.parms` file, which provides a centralized mechanism to administer firmware. However, you can enable the Configuration GUI on a client, which enables users to modify a Sun Ray Client's local configuration. See [Chapter 14, *Sun Ray Client Firmware*](#) for details.

14.3. How to Update Firmware on Sun Ray Clients



Note

This separate procedure is not needed if you install and update the firmware as part of a Sun Ray Software installation.

This procedure shows how to download and install the latest Sun Ray Operating Software (firmware) on a Sun Ray server and update the server's `.parms` files with the new firmware. Once the firmware server is updated, the Sun Ray Clients using that firmware server will download and update the new firmware on their next reboot.

This procedure is provided for Sun Ray servers that have been configured on a shared network (LAN) with external DHCP server support. For alternate network configuration, see the examples after the procedure.

If you need to enable the Configuration GUI on Sun Ray Clients once their firmware is updated, see [Section 14.4, “How to Enable and Disable the Configuration GUI on All Sun Ray Clients”](#).

1. Download and unzip the latest Sun Ray Operating Software and make it accessible to the Sun Ray server.

See <http://www.oracle.com/technetwork/server-storage/sunrayproducts/downloads/index.html>

2. Become superuser on the Sun Ray server.
3. Change directory to the unzipped firmware directory and install the firmware.

```
# ./utfwinstall
```

4. Make the firmware accessible to the Sun Ray Clients:

```
# utfwadm -AaV
```

This command is for a Sun Ray server configured on a shared network (LAN) with external DHCP server support (used `utadm -L on` for network configuration). See the examples following this procedure for alternate network configurations.

5. Power-cycle the Sun Ray Clients to update to the new firmware.



Note

You can also use the `utfwload` command to force a firmware update on all Sun Ray Clients that have an older firmware. The `-l` option forces an update on all clients connected to sessions with no user logged in, and the `-L` option forces an update on all clients connected to sessions.

6. Repeat this procedure on each Sun Ray server in a failover group being used as a firmware server.



Note

To update firmware versions for a specific client, use the `utfwadm -e MAC_address` option.

Alternate Network Configuration Examples

Use the following `utfwadm` command examples for alternate network configurations:

- On a shared network (LAN) with Sun Ray server DHCP support (used `utadm -A subnet` for network configuration)

```
# utfwadm -Aa -N all
```

- On a private network (used `utadm -a intf` for network configuration):

```
# utfwadm -Aa -n all
```

14.4. How to Enable and Disable the Configuration GUI on All Sun Ray Clients

This procedure describes how to enable the Configuration GUI on all Sun Ray Clients, so users can locally configure how the clients initialize and boot. Enabling the Configuration GUI essentially enables the client to use its local configuration values first (if any) when initializing.

In previous versions of the Sun Ray Software 5.2 release, two versions of firmware were shipped: one firmware with GUI capability and another firmware without GUI capability. Enabling the firmware with GUI capability was accomplished by loading the GUI firmware onto a Sun Ray Client.

Now, the two firmware versions are combined into a single version, and additional control mechanisms are provided to enable or disable the Configuration GUI. In order to provide a reasonable migration path from the previous configuration, the Configuration GUI will be enabled automatically if there is any local configuration defined on the Sun Ray Client. The new control mechanisms provide a way for you to force the Configuration GUI on or off, or provide a way for each user to enable the Configuration GUI through the use of a password you define.

Enabling or disabling the Configuration GUI is managed using two new keywords in the `.parms` files, `enablegui` and `disablegui`. In order to provide some security, there are also two new control files, `SunRay.enableGUI` and `SunRay.disableGUI`, that act like keys to unlock enabling or disabling the Configuration GUI. These control files must be installed along with the firmware and `.parms` files, and they must be readable by the managed Sun Ray clients. The `utfwadm` command has options to set these keywords and automatically install the control files, as needed.

Once you enable the Configuration GUI on a Sun Ray Client, you can use the Configuration GUI to update the client's local configuration. See [Section 14.5, "How to Modify a Sun Ray Client's Local Configuration \(Configuration GUI\)"](#) for more information.

1. Become superuser on the Sun Ray server.
2. Enable the Configuration GUI on Sun Ray Clients in a shared network (LAN) configuration with external DHCP support (configured network using `utadm -L on`):

```
# utfwadm -AaV -G GUI-control
```

The options for `GUI-control` are:

<code>off</code>	The Configuration GUI cannot be enabled. This is the default option.
<code>none</code>	Enables the Configuration GUI after using Stop-M or Stop-C on the client. No password is required.
<code>force</code>	Enables the Configuration GUI.
<code>hashed-passwd</code>	The hashed password that the user must enter to enable the Configuration GUI. This option requires you to get the generated hashed password from the <code>uthashpwd</code> command, which takes a password from standard input and prints the hashed result.
<code>prompt</code>	Prompts you to enter the password that the user must enter to enable the Configuration GUI. The password is processed by the <code>uthashpwd</code> command and the resulting value is assigned to the <code>enablegui</code> value.

This command updates the `enablegui` keyword in the `.parms` file. For more details, see the `utfwadm` man page.



Note

The `-g` option disables the Configuration GUI and accepts the same options.

3. Power-cycle the clients to put the new firmware mode into effect.
4. Repeat these steps on each Sun Ray server in a failover group.

**Note**

To enable the Configuration GUI for a specific client, use the `utfwadm -e MAC_address` option.

Alternate Network Configuration Examples

Use the following `utfwadm` command examples for alternate network configurations:

- On a shared network (LAN) with Sun Ray server DHCP support (configured network using `utadm -A subnet`):

```
# utfwadm -Aa -N all -G GUI-control
```

- On a private network (configured network using `utadm -a intf`):

```
# utfwadm -Aa -n all -G GUI-control
```

14.5. How to Modify a Sun Ray Client's Local Configuration (Configuration GUI)

Sun Ray Software provides optional functionality to modify a Sun Ray Client's local configuration through a Graphical User Interface (GUI) tool. A Sun Ray Client's local configuration is checked first before using the configuration from the Sun Ray server, so this enables you to individually configure a Sun Ray Client's behavior at the local level.

Most of the firmware values are stored in the Sun Ray Client's flash memory. Certain control key combinations are used to invoke the Configuration GUI, which enables you to examine and set the local configuration values.

The Configuration GUI enables several features that require the ability to set and store configuration information on the Sun Ray Client itself, including:

- Non-DHCP network configuration for standalone operation, when configuring local DHCP operation is impossible
- Local configuration of Sun Ray specific parameters, such as server list, firmware server, MTU, and bandwidth limits
- DNS servers and domain name for DNS bootstrapping
- VPN configuration
- 802.1x configuration
- IPsec configuration

The firmware server specified in a client's local configuration is the default server used to provide configuration information for download, such as certificate files, `.pcf` files, the `.parms` file, and configuration files.

14.5.1. Security Configuration Repository

A security configuration repository is provided in a Sun Ray Client's firmware to store specific configuration files and certificates/keys for features such as VPN or 802.1x authentication. You can copy files to a firmware's repository through the file copy entry in a remote configuration file. See [Table 14.3, "Remote Configuration File Key Values"](#) for details.

Files stored in the firmware's repository are typed by the directory in which they are placed. The current directories and types are:

- 802.1x authentication
 - `/certs` - X509 certificate files
 - `/keys` - Public/private key files
 - `/wpa` - wpa_supplicant configuration files
- IPsec
 - `/ike/default.conf` - IKE configuration file (racoon configuration file)
 - `/preshared/keys` - Pre-shared key file (used when `authentication_method` statement set to `pre_shared_key`)
- VPN
 - `/profiles` - Cisco VPN configuration profiles (`.pcf` files)

In addition to the files that you copy to the firmware's repository, other files may be created by some configuration operations.

14.5.2. Configuration GUI Menu Descriptions

Table 14.1, "Configuration GUI Main Menu Items" and Table 14.2, "Configuration GUI Advanced Menu Items" provide descriptions for the Configuration GUI menu items.

- Press one of the following key combinations on a Sun Ray Client to open the Configuration GUI and display the main menu:
 - Stop-S or Ctrl-Pause-S
 - Stop-M or Ctrl-Pause-M

Some of the menus have an **Exit** entry, but the Escape key always invokes one level higher than the current menu. Escape at the top level prompts for any changes to be saved or discarded. If changes have been written to the flash memory, the Escape key resets the Sun Ray Client.

Table 14.1. Configuration GUI Main Menu Items

Main Menu Item	Menu Item Descriptions
VPN Setup	Cisco EzVPN authentication model <ul style="list-style-type: none">• Enable - On/Off• Import profile - Profile name• Peer type - Cisco or Netscreen (Juniper Networks)• Auth method - Xauth, Preshared, or Hybrid• Peer - Gateway peer (name or IP address)• Group - Group name• Set Group Key

Main Menu Item	Menu Item Descriptions
	<ul style="list-style-type: none"> • Username - Xauth user name (if static) • Set Password - Xauth password (if static) • Set PIN - If the PIN has been set, the user is prompted for it before a locally stored Xauth user name and password are used. • Advanced <ul style="list-style-type: none"> • DH Group - Diffie-Hellman group • PFS Group • IKE Lifetime - IKE Phase 1 lifetime • IPsec Lifetime • Dead Peer Detection • Session timeout - Idle timeout, after which VPN connection is dropped • Save - Save the VPN configuration.
802.1x Configuration	<ul style="list-style-type: none"> • Enable and initialize - Enables 802.1x authentication. If you choose this menu item and the wired.conf file does not exist, you are prompted to create the file in the Sun Ray Client firmware and the Sun Ray Client reboots if you accept. The reboot is required to complete the 802.1x initialization. After rebooting, choose Configure to add configuration values to the wired.conf file. • Disable - Disable 802.1x authentication. This menu item removes the wired.conf file from the Sun Ray Client firmware. The Sun Ray Client must reboot to complete the process. • Configure - Provides a list of configuration values that can be changed in the wired.conf file. <p>All string values, including file names, need to be enclosed in double quotes, otherwise, they will be parsed as hexadecimal strings. You can specify NULL (without quotes) in a field to represent a variable that has no value and causes a value to be cleared. Selections for file names (keys or certificates) are displayed as a list of the available files of the correct type from the corresponding directories, including the NULL selection.</p> <p>The full description of these values are provided in the wpa_supplicant example configuration file.</p> <ul style="list-style-type: none"> • ssid - SSID (network name). This value is fixed as "wired" and it cannot be changed. • key_mgmt - List of accepted authentication protocols. Values include NONE (no authentication) or IEEE8021X (perform 802.1x using EAP authentication). • eap - List of acceptable Extended Authentication Protocol (EAP) methods. Only one value can be specified. Values include MD5, TLS, MSCHAPV2, PEAP, TTLS, GTC, and OTP. • ca_cert - File path to the certificate file in the /certs directory, with one or more trusted CA certificates, used for EAP-TLS/TTLS/PEAP.

Main Menu Item	Menu Item Descriptions
	<ul style="list-style-type: none"> anonymous_identity - Anonymous identity string for EAP that supports a different tunneled identity, such as EAP-TTLS and EAP-PEAP. If this is defined, it is used as the initial EAP identity, and "identity" is used in any phase 2 protocol. identity - Identity string for EAP password - Password string for EAP. private_key - File path to the client private key file in the <code>/keys</code> directory. (No <code>private_key_passwd</code> needs to be defined, as the private key is stored in the Sun Ray Client flash memory that cannot be accessed.) client_cert - File path to a client certificate file in the <code>/certs</code> directory, for example, for EAP-TLS. phase2 - Inner authentication parameters. This field enables you to specify the internal authentication mode for EAP-PEAP or EAP-TTLS. Example values include <code>"auth=xxx"</code> or <code>"autheap=xxx"</code>, where <code>xxx</code> is the selected inner authentication mode. If this value is not set, then any available authentication mode is allowed. ca_cert2 - File path to certificate file in the <code>/certs</code> directory for use in phase 2 authentication. private_key2 - File path to client private key file in the <code>/keys</code> directory for use in phase 2 authentication. client_cert2 - File path to client certificate file in the <code>/certs</code> directory for use in phase 2 authentication. <p>Note: A certificate with a passphrase is not supported.</p>
VPN Profiles	<ul style="list-style-type: none"> Download Profile File Remove Profile File Show Profiles Clear All Profile Files
Certificates	<ul style="list-style-type: none"> Download Certificate File Remove Certificate File Show Certificates Clear All Certificate Files <p>Note: A certificate with a passphrase is not supported.</p>
Servers	<ul style="list-style-type: none"> Server list - A list of comma-separated server names or IP addresses Firmware server - Name or IP address <pre>[{tftp http}://]server-name-or-IP</pre>

Main Menu Item	Menu Item Descriptions
	<p>Trivial File Transfer Protocol (TFTP) is the default transport and <i>server-name-or-IP</i> specifies the default server used to provide configuration information for download, including certificate files, <i>.pcf</i> files, <i>.parms</i> file, firmware, and configuration files.</p> <p>When using TFTP, the files must be accessible from the server's TFTP home directory. When using HTTP, the files must be located in or linked to the web server's document directory.</p> <ul style="list-style-type: none"> Log host - IP address of syslog host
Network	<ul style="list-style-type: none"> Network configuration - IPv4 (default) or IPv6
TCP/IP	<ul style="list-style-type: none"> Auto (available for IPv6) DHCP - MTU (available for IPv4) Static - IP address, netmask, router, broadcast address, MTU (IPv4) or IP address, Prefix Length, Router, MTU (IPv6)
DNS	<ul style="list-style-type: none"> Domain name - One only DNS server list - List of IP addresses
Authentication	<p>Set if network connection requires a simple HTTP authentication before it can be used.</p> <ul style="list-style-type: none"> Enable/Disable switch Port number
Security	Set password (lock configuration under password control)
Status	Version (equivalent to Stop-V)
Advanced	See below.
Clear Configuration	Equivalent to Stop-C.
Exit	Exit the Configuration GUI.

Table 14.2. Configuration GUI Advanced Menu Items

Main Menu Item	Description
Download Configuration	<p>Prompts for a server name and the file name of a remote configuration file to be downloaded from the server, in the form:</p> <pre>[{tftp http}://][<i>server-name-or-IP</i>]/<i>file-name</i></pre> <p>This field can be overwritten when selected. Pressing Return causes the corresponding remote configuration file to be read and the configuration values parsed and set on the client. For configuration values, see Table 14.3, "Remote Configuration File Key Values".</p> <p>The default transport used is TFTP and the default port is the corresponding port for the transport, 69 for TFTP and 80 for HTTP. The default server is the firmware server value in the local configuration (if <i>server-name-or-IP</i> is not defined) and the default file name is <i>config.MAC</i>, where <i>MAC</i> is the unit's MAC address in upper-case hexadecimal.</p> <p>When using TFTP, the remote configuration file must be accessible from the server's TFTP home directory. When using HTTP, the remote configuration file must be located in or linked to the web server's document directory.</p>

Main Menu Item	Description
Keyboard Country Code	A keyboard country code (keyboard map) that is applied to a keyboard that returns a country code of 0, for use with non-U.S. USB keyboards that do not report a country code. For the list of valid keyboard country code values, see Section 13.2.10, "Keyboard Country Codes" .
Bandwidth Limit	The maximum amount of network bandwidth in bits per second that a given client will use.
Session Disconnect (Stop-Q)	Enables or disables the ability to terminate a session by pressing Stop-Q. This feature is useful when you want to terminate a VPN connection and leave the Sun Ray in an inactive state. Pressing the Escape key after the session has terminated reboots the Sun Ray Client.
Force Compression	Sets a tag sent from the Sun Ray Client to the Xserver telling it to enable compression regardless of available bandwidth.
Lossless Compression	Disables the use of lossy compression for image data.
Disallow utload	Disables the ability to explicitly force a firmware load into a Sun Ray Client. In this way, firmware can be tightly controlled using .parms files or DHCP parameters.
Force Full Duplex	Allows the Sun Ray Client to operate correctly when the network port that it is connected to does not auto-negotiate. In that case, the auto-negotiation results in the Sun Ray running at half duplex, which significantly impacts network performance. This setting allows the Sun Ray to operate with better performance in this situation.
Enable Fast Download	<p>If set, the Sun Ray Client uses the maximum TFTP transfer size if the TFTP server supports it. Over a high latency connection, this setting typically doubles the speed of firmware downloads. There are no disadvantages to enabling fast downloads on low latency LANs.</p> <p>This parameter is disabled by default and the transfer size is set at 512-byte packets. It is disabled by default for backwards compatibility with TFTP servers that might not support the more advanced protocol. If this parameter were on by default and a firmware download were to fail, there would be no way to recover.</p>
Power Off Timer	Energy star power off feature for Sun Ray 3 Series Clients. The value for the power off feature is in minutes. The default power off time is 30 minutes. A value of 0 disables the power off feature.
Enter Alternate STOP modifiers	<p>Specifies an alternative combination of modifier keys to perform the same function as the Stop key on an Oracle keyboard or the Ctrl-Pause key sequence on a non-Oracle keyboard. By default, this alternative combination is Ctrl-Shift-Alt-Meta. See Section 13.2.2, "Sun Ray Client Hot Keys" for details.</p> <p>You can change Ctrl-Shift-Alt-Meta to any other combination of the same keys, but at least two of the keys must be used. For example, you can set this value to Ctrl-Alt or Meta-Ctrl-Shift.</p> <p>If this parameter is set to none, the alternative key combination is disabled.</p> <p>Note that the Meta key has different names on different keyboards: on a PC keyboard, it is the "Windows" key, and on a Mac keyboard, it is the "Command" key.</p>
Command Cache Size	Specifies the size, in Kbytes, of the command cache look-back buffer. This area is used to store a list of recent commands used by the firmware, and the commands are replayed from the cache if used again. The default value is 512 Kbytes, maximum value is 8192 Kbytes, and a zero value disables the command cache.
Video	<ul style="list-style-type: none"> Blanking - Specifies the blanking timeout, which is the time until the screen is put to sleep, in minutes. Specify 0 to disable.

Main Menu Item	Description
Video input disable	Sun Ray 270 Client only. If set, turns off the input selector on the front of the client and locks the monitor so that it displays only the Sun Ray output. This feature prevents users from connecting a PC to the VGA video input connector on a client and using it as a monitor.

14.5.3. How to Load a Remote Configuration File

To help avoid error-prone manual entry of local configuration data or to help configure a lot of Sun Ray Clients more quickly, you can use the [Download Configuration menu](#) item to download a pre-defined remote configuration file from a server via TFTP or HTTP.

The keywords shown in [Table 14.3, “Remote Configuration File Key Values”](#) correspond to configuration values that can be set from the Configuration GUI menus. To group items that are logically related, some of the keywords take the form *family.field*.

Table 14.3. Remote Configuration File Key Values

Key Values	Description
<i>target-file-path=file-to-copy</i> (file copy entry)	You can copy configuration files and certificates/keys to the firmware's security configuration repository by using a file copy entry. A file copy entry follows the normal <i>key=value</i> format, except the <i>key</i> used is the absolute path name of the target file and it must begin with a "/" character. The <i>value</i> used is the configuration file to be copied, which needs to be located in the same location as the remote configuration file. You can use the file copy entries for both VPN and wpa_supplicant configuration files. For example, the file copy entry <i>/wpa/wired.conf=wired_config</i> will copy the file <i>wired_config</i> from the configuration server to the <i>/wpa/wired.conf</i> file on the Sun Ray Client. Once you add all the necessary file copy entries, you can choose Advanced > Download Configuration in the Configuration GUI to download the remote configuration file and copy the files specified. See Section 14.5.1, “Security Configuration Repository” for more information.
VPN/IPsec Submenu	
vpn.enabled	Enable toggle
vpn.peer	Remote gateway name/IP address
vpn.group	VPN group
vpn.key	VPN key
vpn.user	Xauth user
vpn.passwd	Xauth password
vpn.pin	PIN lock for use of user/passwd
vpn.peertype	Cisco or Netscreen
vpn.authtype	Xauth, Preshared, or Hybrid
vpn.dhgroup	Diffie-Hellman group to use
vpn.pfsgroup	PFS group to use
vpn.lifetime	Lifetime of IKE connection
vpn.ipseclifetime	Lifetime of IPsec connection
vpn.dpdswitch	Dead peer detection
vpn.killtime	Idle timeout value to drop VPN connection.
DNS Submenu	

Key Values	Description
dns.domain	Domain name
dns.servers	Server list (comma-separated IP addresses)
Servers Submenu	
servers	Sun Ray server
tftpserver	Firmware (TFTP) server
loghost	Syslog host
Security Submenu	
password	Set administrator password
Network Submenu	
network	Type of network (IPv4 or IPv6)
TCP/IP Submenu	
ip.ip	Static IPv4 address
ip.mask	Static netmask
ip.bcast	Static broadcast address
ip.router	Static router
ip.mtu	MTU
ip.type	IP address source (DHCP or Static)
TCP/IPv6 Submenu	
ip.ip6	Static IPv6 address
ip.prefix	Static IPv6 prefix
ip.router	Static router
ip.mtu	MTU
ip.type	IP address source (Auto or Static)
Advanced Submenu	
kbcountry	Keyboard country code
bandwidth	Bandwidth limit in bits per second.
stopqon	Enable (1) or Disable (0) Stop-Q for disconnect
compress	Force compression on when 1
lossless	Force use of lossless compression when 1
utloadoff	Disallow use of utload to force firmware download when 1
fastload	Force maximum TFTP transfer rate when 1
fulldup	Force full-duplex when 1
poweroff	Poweroff time in minutes
stopkeys	Change alternate combination of keys used for Stop key
cmdcachesize	Command cache size
videoindisable	Disable input selector of Sun Ray 270 Client when 1

The format of the file is a set of *key=value* lines, each terminated by a newline character, which are parsed and the corresponding configuration items set (see the sample file below). No whitespace is

permitted. Key values are case-sensitive and should be always lower case, as listed above. Setting a keyword to have a null value results in the configuration value being cleared in the local configuration.

14.5.3.1. Sample VPN Configuration File

```
vpn.enabled=1
vpn.peer=vpn-gateway.company.com
vpn.group=homesunray
vpn.key=abcbabcabc
vpn.user=johndoe
vpn.passwd=xyzxyzxyzzy
dns.domain=company.com
tftpserver=config-server.company.com
servers=sunray3,sunray4,sunray2
```

14.6. VPN Support

Sun Ray Clients are able to provide a VPN solution for remote users. The IPsec capability in the Sun Ray Client firmware enables the Sun Ray Client to act as a VPN endpoint device. The most commonly used encryption, authentication, and key exchange mechanisms are supported, along with Cisco extensions that enable a Sun Ray Client to interoperate with Cisco gateways that support the Cisco EzVPN protocol. Sun Ray Clients currently support IPsec VPN concentrators from Cisco and Netscreen (Juniper).

The security model is identical to that of the Cisco software VPN client. Using a common group name and key for the initial IKE phase one authentication exchange, the client authenticates the user individually with the Cisco Xauth protocol, either by presenting a fixed user name and password stored in flash memory or by requiring the entry of a user name and one-time password generated by a token card.

VPN support relies on the Configuration GUI and the following implementations are supported:

- Cisco Hybrid authentication
- Certificates using OpenSSL
- Cisco PSK hash for Hybrid authentication
- Cisco gateway load balancing redirection from first connection
- Gateway setting that allows or disallows stored passwords
- Cisco Profile Configuration File ([.pcf](#)) files, including decrypting the encrypted group password
- Ability to load certificates and [.pcf](#) files over the network
- Gateway setting for Perfect Forward Secrecy (PFS)
- Configuration of authentication mode (preshared, Hybrid, or XAUTH)
- IKE fragmentation for large negotiation packets

To protect the use of stored authentication information, the VPN configuration includes a PIN entry. This feature enables two-factor authentication for Sun Ray at Home VPN deployments.



Note

You can also copy VPN configuration and certificate files to the firmware by using the file copy entry in a remote configuration file. See [Table 14.3, “Remote Configuration File Key Values”](#) for details.

14.6.1. How to Configure VPN Using Cisco Hybrid Authentication

This procedure describes the steps to modify the Configuration GUI to use Cisco Hybrid authentication.

The procedure assumes that the Sun Ray Client has access to an appropriate server supplying the necessary configuration files.

1. Press Stop-M or Ctrl-Pause-M to open the Configuration GUI.
2. Choose **Certificates > Load Certificate File**.
 - Enter the URL of a file containing the root certificate in PEM format, which is used to sign the gateway certificates.
 - Exit the menu.
3. Choose **VPN Profiles > Load Profile File**.
 - Load any appropriate Cisco `.pcf` files.
 - Exit the menu.
4. Choose **VPN Setup > Import VPN profile**.
 - Cycle through the existing `.pcf` files by hitting Enter until the desired profile is selected. The values from this file will be populated into the submenu entries, but they will not be stored until the values are saved. Cycling back to the initial entry with no `.pcf` file selected will restore the initial values.
5. Set more values in the VPN-Setup menu.
 - Set Enable to on.
 - Set any other VPN values desired, such as Username.
 - Save the VPN settings.
6. (Optional) Choose **Advanced > Download Configuration** to download the VPN settings

The new Auth method is specified in the configuration file as "vpn.authmethod", and the valid values are case-insensitive "xauth", "preshared", and "hybrid".
7. Enter ESC from the main menu and save the Configuration GUI settings.

The Sun Ray Client will reboot and try to make the VPN connection.

14.7. IPsec

Beyond the IPsec capability to make Sun Ray Client as a VPN endpoint device, Sun Ray Software also supports IPsec to provide high quality, cryptographically-based security between Sun Ray Clients and Sun Ray servers. After configuring and enabling IPsec on the Sun Ray server and the Sun Ray Client, the Sun Ray Client will negotiate a secure end-to-end IPsec tunnel with the Sun Ray server before interacting with Sun Ray services on the server.

The Sun Ray Software implementation of IPsec is incorporated into the Sun Ray Client firmware. The Sun Ray Client will always be the initiator of a connection, so it does not have to respond to inbound connection requests. This type of negotiation is similar to the current IPsec VPN behavior, where IPsec is established with a VPN gateway before Sun Ray services are invoked. However, both IPsec implementations require different configurations.

See [Appendix A, IPsec Support](#) for details.

14.8. 802.1x Authentication

The 802.1x authentication feature in the Sun Ray Client firmware is based on an Open Source project called `wpa_supplicant`, which is described at http://hostap.epitest.fi/wpa_supplicant/. With the 802.1x authentication feature, Sun Ray Clients can be configured to provide proper credentials to successfully authenticate and gain access to the local area network under 802.1x access control. Sun Ray Clients support the Extensible Authentication Protocol Modes: MD5, TLS, MSCHAPV2, PEAP, TTLS, GTC, and OTP.

`wpa_supplicant` supports the implementation of the WPA supplicant protocol for wireless authentication, which includes the 802.1x port authentication protocol. As a result, the configuration of 802.1x depends on the mechanisms and configuration file format provided by `wpa_supplicant`.



Note

Although the WPA supplicant protocol is primarily targeted for wireless authentication, Sun Ray Clients do not currently supported wireless operation.

`wpa_supplicant` uses a main configuration file to configure the 802.1x authentication, along with a few secondary files containing certificates and public/private key pairs. The main configuration file used with the Sun Ray Software is named `wired.conf`. In order for `wpa_supplicant` to access the configuration files, you need to copy them to the Sun Ray Client firmware's security configuration repository by using file copy entries in a remote configuration file. See [Table 14.3, "Remote Configuration File Key Values"](#) for details.

The `wired.conf` file must be present on a Sun Ray Client in order to start the `wpa_supplicant` component and to attempt 802.1x authentication. The presence or absence of this configuration file is the primary mechanism used to enable or disable `wpa_supplicant`. The 802.1x Configuration menu item in the Configuration GUI enables you to manage the `wired.conf` file, which uses only a reduced set of configuration values required for various authentication modes of 802.1x. The configuration options are further refined depending on the particular Extended Authentication Protocol (EAP) mode selected. See [Table 14.1, "Configuration GUI Main Menu Items"](#) for details.

Currently, private keys cannot be generated on the Sun Ray Client itself, so you must generate the private keys and corresponding certificates by other means and provide them through the remote configuration file.

If you create and modify the `wired.conf` file outside of the Configuration GUI, make sure the appropriate fields are provided and the file is formatted correctly. The file must have the single network definition of `ssid="wired"` included. If the `wired.conf` file does not follow the expected format, `wpa_supplicant` will fail to operate correctly. See the contents of the `wired.conf` file in the following example.

14.8.1. How to Configure and Enable 802.1x Authentication on a Sun Ray Client

This procedure describes how to configure and enable 802.1x authentication on a Sun Ray Client. The steps include examples to set up an 802.1x authentication using the EAP-TLS mode of operation.



Note

The configuration files listed in the procedure must be available in the same location as the remote configuration file, which is usually the firmware server defined in the local configuration.

1. Create the configuration files for wpa_supplicant, including the main configuration file, `wired.conf`, and the secondary files containing certificates and public/private keys.

For the list of valid `wired.conf` values, see the 802.1x Configuration menu descriptions in [Table 14.1, "Configuration GUI Main Menu Items"](#).

Here are some examples of secondary files and the `wired.conf` file.

`someca_cert.pem` - a Certificate Authority root certificate from "someca"

```
-----BEGIN CERTIFICATE-----
MIID3DCCA0WgAwIBAgIBADANBgkqhkiG9w0BAQUFADCBOzETMBEGCgmSJomT8ixk
ARkWA2NvbTETMBEGCgmSJomT8ixkARkWA3N1bjEVMBMGCGmSJomT8ixkARkWBXNm
....
CkS0he0fm5xVRd6D+nQQAbUkFy0MZO39QjXbopBxaY5Vm5hg2U+00JJ5UHQXGGMk
sxyGuzhrnu09oYF7ZjelBlO2fGhC/JrSJhKFQtggNBQ=
-----END CERTIFICATE-----
```

`sunray_key.pem` - a RSA key pair for the Sun Ray Client

```
-----BEGIN RSA PRIVATE KEY-----
MIICXAIBAAKBgQCvGwBJjv/Uzp8lQAd9B9uqehZqmS9BVA9xcfJtNf6Feou3FnKE
8tHcCISAXFdujYzSghzcInzn/ZWnKk2cRQl8//IupuMcwPil0QebBmXhxfrrTTW5L
....
FEmkooUWfa6mUpAcpQJBANCe64twQ3RjNfIc3n4LpCEPgW7y5pgk8xmKIDiSZ/+U
XwJQ4gpzmsakaZWBECdxrJWkK6chvcFwcwcfAN7rkOBc=
-----END RSA PRIVATE KEY-----
```

`sunray_cert.pem` - a client certificate for the Sun Ray Client RSA key, signed by "someca"

```
-----BEGIN CERTIFICATE-----
MIIE+TCCBGKgAwIBAgIBCTANBgkqhkiG9w0BAQUFADCBOzETMBEGCgmSJomT8ixk
ARkWA2NvbTETMBEGCgmSJomT8ixkARkWA3N1bjEVMBMGCGmSJomT8ixkARkWBXNm
....
vv7TQOt1SlwPessnDJOFJ+oYoAMbc3f8bmVOMvqQ98zZGdJ/VDK+siFJKeTpkol
ocRIJUfegNu4W0+pvgPY/ZBsbUchBA2rpdhwWnc=
-----END CERTIFICATE-----
```

`wired.conf` - wpa_supplicant configuration file for 802.1x/EAP-TLS

```
network={
    ssid="wired"
    key_mgmt=IEEE8021X
    eap=TLS
    ca_cert="/certs/someca.pem"
    identity="john.doe@oracle.com"
    private_key="/keys/sunray.pem"
    client_cert="/certs/sunray.pem"
}
```

2. Create a remote configuration file with the needed file assignment entries, which will be used to copy the wpa_supplicant configuration files to the Sun Ray Client.

Here is an example of a remote configuration file:

```
/certs/someca.pem=someca_cert.pem
/keys/sunray.pem=sunray_key.pem
/certs/sunray.pem=sunray_cert.pem
/wpa/wired.conf=wired.conf
```

The `/wpa/wired.conf=wired.conf` entry is required.

3. Download the remote configuration file to a Sun Ray Client by choosing **Advanced > Download Configuration** in the Configuration GUI.

Once the `wired.conf` file is loaded, 802.1x authentication is automatically enabled if the `key_mgmt` key is set to `IEEE8021X`.

4. (Optional) Make changes to the `wired.conf` file by choosing 802.1x Configuration in the Configuration GUI.
5. Plug the Sun Ray Client into a port that provides 802.1x authentication and test the authentication.

See [Section 16.14, “\(20\) 802.1x Authentication Icon”](#) for information about possible error codes or status messages.

14.9. How to Display Firmware Versions for All Currently Connected Sun Ray Clients

1. Become superuser on the Sun Ray server
2. Display the firmware versions.

```
# /opt/SUNWut/sbin/utfwload -a
```

You can also use the `utquery -d` command.

14.10. How to Display the Firmware Version from a Sun Ray Client

Press Stop-V or Ctrl-Pause-V.

14.11. How to Synchronize the Sun Ray Client Firmware

This procedure uses the `utfwsync` command to synchronize the currently installed and configured firmware on a Sun Ray server with all the other Sun Ray servers in its failover group and to enable firmware updates to occur on the Sun Ray Clients. This command also forces a reboot on all the Sun Ray Clients to update to the new firmware if needed. The `utfwsync` command is primarily used during a Sun Ray Software upgrade

1. Become superuser on the Sun Ray server.
2. Synchronize the Sun Ray Client firmware.

```
# /opt/SUNWut/sbin/utfwsync
```

The Sun Ray Clients reboot themselves and update to the new firmware if needed.

14.12. How to Downgrade Firmware on a Sun Ray Client

As a general recommendation, your Sun Ray Clients should always have the latest firmware installed. However, there may be times when you need to downgrade the firmware to a previous release.

Every firmware has its own barrier level value, which is used to determine if the firmware installed on the Sun Ray server is newer than the one installed on the Sun Ray Client as it boots up. In most cases, if the firmware barrier level on the Sun Ray Client is lower than the firmware barrier level on the Sun Ray server, the new firmware will be automatically updated on the Sun Ray Client when it boots.

To downgrade firmware on a Sun Ray Client, you can use the `-F` option of the `utfwadm` command to manipulate the barrier level and force the downgrade to occur. When using this option, the `BarrierLevel` key is set in the client-specific `.parms` entry.

Use the following procedure to downgrade the firmware on a specific Sun Ray Client.

1. Download and unzip the Sun Ray Software media pack containing the firmware you want to install and make it accessible to the Sun Ray server.

**Note**

In Sun Ray Software releases before the 5.3 release, the Sun Ray Client firmware images were part of the Sun Ray Software media pack. Starting with the Sun Ray Software 5.3, the firmware images were separated into the Sun Ray Operating Software release.

See <http://www.oracle.com/technetwork/server-storage/sunrayproducts/downloads/index.html>

2. Become superuser on the Sun Ray server.
3. Configure the firmware downgrade for a Sun Ray Client.

```
# utfwadm -AVF -e enetAddr -f firmware
```

Where *enetAddr* is the MAC address of the Sun Ray Client to downgrade and *firmware* is the relative path to the firmware image. The firmware image is located in the firmware package in the Sun Ray Software media pack. For example, in the Sun Ray Software 5.2 media pack, it is located in the `srss_4.3/Components/10-SRSS/Content/Sun_Ray_Core_Services_4.3/Solaris_10+/sparc/Packages/SUNWutfw/reloc/SUNWut/lib/firmware` directory.

This command is for a Sun Ray server configured on a shared network (LAN) with external DHCP server support (used `utadm -L on` for network configuration). See the examples following this procedure for alternate network configurations.

4. Power-cycle the Sun Ray Client to downgrade the firmware.

**Note**

If you want to update the Sun Ray Client back to the latest firmware installed on the Sun Ray server, use the following command on the server to remove the specific client's entry from the `.parms` file and power-cycle the Sun Ray Client:

```
# utfwadm -DV -e enetAddr
```

Alternate Network Configuration Examples

Use the following `utfwadm` command examples for alternate network configurations:

- On a shared network (LAN) with Sun Ray server DHCP support (used `utadm -A subnet` for network configuration)

```
# utfwadm -AF -f firmware -e enetAddr
```

- On a private network (used `utadm -a intf` for network configuration):

```
# utfwadm -AF -f firmware -e enetAddr
```


14.13. How to Disable All Sun Ray Client Firmware Updates

This procedure is needed when upgrading Sun Ray Software on Sun Ray servers in a failover group.

1. Become superuser on the Sun Ray server.
2. Disable all firmware updates.

For a shared network (LAN) with external DHCP server support (used `utadm -L on` for network configuration)

```
# ./utfwadm -D -a -V
```

For a shared network (LAN) with Sun Ray server DHCP support (used `utadm -A subnet` for network configuration)

```
# ./utfwadm -D -a -N all
```

For a private network (used `utadm -a intf` for network configuration)

```
# ./utfwadm -D -a -n all
```

Chapter 15. Peripherals

Table of Contents

15.1. Peripherals Overview	197
15.2. Enabling and Disabling Device Services	198
15.2.1. How to Determine the Current State of Device Services	198
15.2.2. How to Enable or Disable USB Device Services	199
15.3. Device Availability Per Session	199
15.4. Accessing Serial Devices and USB Printers	199
15.4.1. Device Links	199
15.4.2. Device Nodes	200
15.4.3. Device Node Ownership	201
15.4.4. Hotdesking and Device Node Ownership	201
15.4.5. Setting Up Serial Devices	201
15.4.6. Setting Up USB Printers	201
15.5. Accessing USB Mass Storage Devices	204
15.5.1. Device Nodes and Links (Oracle Solaris)	204
15.5.2. Device Nodes and Links (Oracle Linux)	204
15.5.3. Mount Points	204
15.5.4. Device Ownership and Hotdesking	205
15.5.5. Mass Storage Devices and Idle Sessions	205
15.5.6. Commands for Common Disk Operation (Oracle Solaris)	205
15.5.7. Commands for Common Disk Operation (Oracle Linux)	206
15.5.8. How to Unmount a Mass Storage Device From a Client	206
15.5.9. Troubleshooting Mass Storage Devices	207
15.6. USB Headsets	207
15.6.1. Tested USB Headsets	207
15.6.2. Tested Applications	208
15.6.3. Additional Notes	208
15.7. USB Device Operations Failing After Idle Timeout Limit	208

This chapter describes how to access peripherals or devices connected to a Sun Ray Client's USB and serial ports or connected to an Oracle Virtual Desktop Client's serial ports (when on a Windows-based client computer).

Most of this chapter pertains to Sun Ray Clients. To find out how to access external peripherals when using an Oracle Virtual Desktop Client, see [Section 13.3.2, "Using External Devices on the Client Computer"](#).

15.1. Peripherals Overview

Sun Ray Software works with various mass storage USB devices such as flash disks, memory card readers, zip drives, and disk drives on Sun Ray Clients. Data CDs and DVDs can be read but not written. Other end-user peripherals such as USB headsets, printers, and serial devices can also be used.

There are multiple ways to access peripherals connected to a Sun Ray Client, based on the type of Sun Ray session you are using:

- **Oracle Linux or Oracle Solaris Sessions** - You can access the device through the device links and nodes automatically created on the Sun Ray server. See [Section 15.4, "Accessing Serial Devices and USB Printers"](#) for details. USB mass storage devices are automatically mounted, which is described in [Section 15.5, "Accessing USB Mass Storage Devices"](#).

USB-to-serial adapters are not accessible through the generated device nodes. You must use USB redirection in a Windows session to access a serial device connected through a USB-to-serial adapter.

- **Windows Sessions Using USB Redirection** - USB redirection is the recommended way to access USB devices connected to Sun Ray Clients when using Windows sessions. USB device redirection requires that the USB device redirection component is installed on the Windows system. See [Section 17.6, “USB Device Redirection”](#) for details. Although most of the information in this chapter is not applicable when using USB redirection, device services still need to be enabled.
- **Windows Sessions Using Logical Device Mapping** - Without USB redirection, you can use the `-r` option of the `uttsc` command to create a logical device mapping to the device being managed on the Sun Ray server, as described in this chapter. For example, the `-r disk:drive=path` enables you to create a logical device mapping to a USB mass storage device, where `path` is the mounted disk located at `/tmp/SUNWut/mnt/user` on the Sun Ray server. Other devices, such as serial devices, can also be configured. For details about accessing serial devices when using the Windows connector, see [Section 17.17, “Accessing Serial Devices”](#).

Using logical device mapping with USB mass storage devices provides much lower performance on Oracle Linux than Oracle Solaris due to the design of the Linux mass storage subsystem. For Windows sessions, use USB redirection for optimum performance with mass storage devices.

For the latest list of peripherals tested to work with Sun Ray Software, see the [Sun Ray Client and Oracle Virtual Desktop Client Peripherals](#) document.

**Note**

Once the latest firmware is installed on a Sun Ray Client, no special installation is required for USB headsets to work. Just connect the USB headset to the Sun Ray Client. See [Section 15.6, “USB Headsets”](#) for details.

15.2. Enabling and Disabling Device Services

Before devices can be accessed on a Sun Ray Client, Sun Ray device services for the device must be enabled. Sun Ray device services can be enabled and disabled with the `utdevadm` command line tool or from the Security tab on the Admin GUI Advanced tab. Sun Ray device services include USB devices connected through USB ports, internal serial ports, and internal smart card readers on the Sun Ray Client.

Here are the available device services for external peripheral devices:

- Internal serial port - Enables users to access the embedded serial ports on their Sun Ray Clients.
- USB port - Enables users to access any devices connected to USB ports. This situation does not affect HID devices such as the keyboard, mouse, or barcode reader.

After you install Sun Ray Software, all device services are enabled by default. You can use the `utdevadm` command to enable or disable device services only in the configured mode, that is, after the Sun Ray Data store is activated.

This configuration affects all the servers in a group and all the clients connected to that group.

15.2.1. How to Determine the Current State of Device Services

The `utdevadm` command displays the enabled or disabled state of device services.

```
# utdevadm
```

15.2.2. How to Enable or Disable USB Device Services

- To enable USB services, use the `utdevadm` command.

```
# utdevadm -e -s usb
```

- To disable USB services, use the `utdevadm` command.

```
# utdevadm -d -s usb
```

15.3. Device Availability Per Session

For every Sun Ray Client or Oracle Virtual Desktop Client session on a Sun Ray server, Sun Ray Software creates a subdirectory under `/tmp/SUNWut/units`. The subdirectory matches the name of the client's identifier, or CID. A Sun Ray Client's CID is named `IEEE802.MACID`, where `MACID` is the Sun Ray Client's MAC Address. An Oracle Virtual Desktop Client's CID is named `MD5.CLIENTID` where `CLIENTID` is the hexadecimal representation of each Oracle Virtual Desktop Client profile's MD5 hash key. For more information on the difference between Sun Ray and Oracle Virtual Desktop Client CIDs, see [Section 13.1.1, "Client ID Differences Between Oracle Virtual Desktop Clients and Sun Ray Clients"](#).

Each session has a `$UTDEVROOT` environment variable, which is an alias to that session's current CID subdirectory. `$UTDEVROOT` is a dynamic variable that updates whenever the user hotdesks between devices. If the `$UTDEVROOT` variable is not available, such as in kiosk mode, you can always display the client ID as described in [Section 13.1.1.1, "How to Display Client ID Information"](#).

The following example shows how to find the CID subdirectory for a Sun Ray Client through the `$UTDEVROOT` variable (using bash shell):

```
# echo $UTDEVROOT
/tmp/SUNWut/sessions/4/unit
# cd -P $UTDEVROOT
# pwd
/tmp/SUNWut/units/IEEE802.0003badc1b9d
```

The subdirectory for each client contains `dev` and `devices` directories. The Sun Ray `dev` directory contains a representation of the logical topology of the devices connected to the client. The `devices` directory contains a representation of the physical topology of some of the devices connected to the client.

When accessing or referencing devices connected to the current client, always use the `dev` directory, because the logical device gets updated whenever a user moves from client to client and the path to the device does not contain any special characters that may need special handling in a script.



Note

Sun Ray Software does not create device nodes for every USB device. Some USB device drivers export their device interfaces through other mechanisms than a traditional UNIX device node.

15.4. Accessing Serial Devices and USB Printers

This section provides information about how to manage serial devices and USB printers connected to a client.

15.4.1. Device Links

Device links are created under the `dev` directory. A link to each serial node is created in `dev/term`, and a link to each locally attached printer is created in `dev/printers`.

Typical device links are:

```
/tmp/SUNWut/units/IEEE802.080020cf428a/dev/term/manufacturer_name-67a
/tmp/SUNWut/units/IEEE802.080020cf428a/dev/printers/1608b-64
```

The device link for the first example is *manufacturer_name-serial_numberindex*, where *index* is an increasing alphabetical character, starting at *a*. If the manufacturer name is not available, the vendor and product ID numbers are used for the name of the device link.

Here is an example of the *dev* directory from a Sun Ray 3 Client, which has an onboard serial port.

```
# cd $UTDEVROOT/dev
# pwd
/tmp/SUNWut/units/IEEE802.002128587259/dev
# ls
term
# cd $UTDEVROOT/dev/term
# pwd
/tmp/SUNWut/units/IEEE802.002128587259/dev/term
# ls -l
lrwxrwxrwx 1 root root 22 Jul 28 17:23 a -> ../../devices/serial:a
```

15.4.2. Device Nodes



Note

USB-to-serial adapters are not accessible through the generated device nodes in an Oracle Solaris or Oracle Linux session. You must use USB redirection in a Windows session to access a serial device connected through a USB-to-serial adapter.

In the *devices* directory, device nodes are created for each serial or printer port on an attached USB device. The device nodes are created in the hub directory corresponding to the hub to which they are attached. The nodes are named as follows:

```
manufacturer_name,model_name@upstream_hub_port
```

If the USB device has multiple identical ports (for example, two serial ports), the name is followed by *:n* where *n* is a numerical index, starting at *1*.

The following example is a typical device node:

```
/tmp/SUNWut/units/IEEE802.MACID/devices/usb@1/hub@1/manufacturer_name,model_name@3:1
```

Here are the definitions of the naming conventions.

Term	Definition
<i>physical-topology</i>	The <i>physical-topology</i> is <i>hub@port/hub@port</i> and so on. The <i>port</i> refers to the port on the parent hub into which the device or child hub is plugged.
<i>printer-name-1, terminal-name-1</i>	The printer and terminal name in the Sun Ray <i>devices</i> directory is <i>manufacturer,model@port</i> with a colon separating the numerical index when the string just described is not unique in the directory.
<i>printer-name-2, terminal-name-2</i>	The printer and terminal name in the Sun Ray <i>dev</i> directory is the manufacturer and serial number concatenated with an alphabetic index when the serial number is not unique.

15.4.3. Device Node Ownership

Some device nodes are owned by the user whose session is active on the client, while others might be owned by root or by other users that had previously active sessions on the client. Device permissions, access controls and ownership rules are determined by the class of device. For serial devices, only the user whose session is active on the client or the superuser have permission to use the attached device. If no user has an active session, superuser owns the serial device nodes. This rule might not be applicable for other classes of USB devices connected to the client.

15.4.4. Hotdesking and Device Node Ownership

The following description of the behavior of USB devices when sessions are connected and disconnected from a client applies only to serial devices. Other device classes may have different semantics regarding ownership and device lease times.

Changing the active session on a client changes the ownership of the device nodes to the user associated with the new session. A session change occurs whenever a user inserts or removes a smart card from a client or logs into a session.

In a failover environment, you can use the `utselect` or `utswitch` command to change a session. A session change causes all devices currently open by a non-root user to be closed after 15 seconds. Any input to or output from any affected device results in an error. For a serial device node, if the original session is restored within 15 seconds, the ownership is not relinquished, and input and output continue uninterrupted.

Devices currently opened by the superuser, including normal printing, remain unaffected by a session change.

15.4.5. Setting Up Serial Devices

To use serial attached devices with a client, you must attach them to the internal serial ports or through a USB-to-serial adaptor. You must use USB redirection in a Windows session to access a serial device connected through a USB-to-serial adaptor.

Symbolic links to the serial port device nodes are located under `$UTDEVROOT/dev/term`. Built-in ports are named "a" or "b".

Serial ports become unowned during hotdesking, so you should make sure any serial port activity is stopped before removing your smart card or resetting the client.



Note

All serial ports except port A on the Sun Ray 170 support full handshaking and standard UNIX semantics. Port A on the Sun Ray 170 has no hardware handshaking pins, so it can't be used when a hardware handshake is required.

15.4.6. Setting Up USB Printers

This section provides instructions on how to setup PostScript and non-attached PostScript printers that are attached to the Sun Ray Client. For details on how to print from Windows while using the Windows connector, see [Chapter 17, Windows Connector](#).

15.4.6.1. How to Set Up an Attached PostScript Printer (Oracle Solaris)

Sun Ray Software works with PostScript printers connected directly to a USB port on the Sun Ray Client. For non-PostScript printer support, refer to [Section 15.4.6.3, "How to Set Up an Attached Non-PostScript Printer"](#).

**Note**

The printer naming conventions in Sun Ray Software differ from those in an Oracle Solaris operating environment.

**Note**

The `lp` subsystem opens the device node as superuser for each print request, so print jobs are not affected by hotdesking.

Starting a print queue on a printer attached to a Sun Ray Client, either directly or through an adapter, is the same process as starting a print queue in Oracle Solaris.

1. On the Sun Ray Client where the printer is attached, log in to a new session as superuser (root).
2. To determine the MAC address of the Client, press the three audio option keys to the left of the power key in the upper right corner of the keyboard.

The alphanumeric string displayed below the connection icon is the MAC address.

3. To locate the Sun Ray Client, type:

```
# cd /tmp/SUNWut/units/*MAC_address
# pwd
/tmp/SUNWut/units/IEEE802.MACID
```

The path to the extended MAC address for your particular Sun Ray Client is displayed.

4. Locate the port for the printer by typing:

```
# cd dev/printers
# pwd
/tmp/SUNWut/units/IEEE802.MACID/dev/printers
# ls
printer-node-name
```

5. In the directory, locate the printer node.
6. Add the new printer.
 - a. Start the Oracle Solaris Print Manager.

```
# /usr/sbin/printmgr &
```

- b. Click **OK** to choose files for repository.
- c. Go to **Printer > New Attached Printer**.
- d. Type the following information:

- Printer name: *prntername*
- Description (optional)
- Printer port
- Printer make
- Printer model

Choose Other to type the printer port path name. To locate the printer port, refer to Step 4.

7. Verify that the printer has been set up correctly.

```
# lpstat -d printername
```

15.4.6.2. How to Set Up an Attached PostScript Printer (Oracle Linux)

Sun Ray Software works with PostScript printers connected directly to a USB port on the Sun Ray Client. For non-PostScript printer support, refer to [Section 15.4.6.3, “How to Set Up an Attached Non-PostScript Printer”](#).



Note

The `lp` subsystem opens the device node as superuser for each print request, so print jobs are not affected by hotdesking.

The following generic instructions might vary slightly from one operating system implementation to another, but they should provide enough information to enable an administrator to set up basic printing services.

1. On the Sun Ray Client where the printer is attached, log in to a new session as superuser (root).
2. To determine the MAC address of the client, press the three audio option keys to the left of the power key in the upper right corner of the keyboard.

The alphanumeric string displayed below the connection icon is the MAC address.

3. Locate the Sun Ray Client.

```
# cd /tmp/SUNWut/units/*MAC_address
# pwd
/tmp/SUNWut/units/IEEE802.MACID
```

The path to the extended MAC address for your particular Sun Ray Client is displayed.

4. Locate the port for the printer.

```
# cd dev/printers
# pwd
/tmp/SUNWut/units/IEEE802.MACID/dev/printers
# ls
printer-node-name
```

5. In the directory, locate the printer node.
6. Use the Oracle Linux administration tools to set up the printer.

Choose Other so that you can provide the device node from Step 4.

7. Verify that the printer has been set up correctly.

```
# lpstat -d printername
```

8. Create a soft link to the Sun Ray printer node in `/dev/usb`

For example, if the device node is

`/tmp/SUNWut/units/IEEE802.mac-address/dev/printers/device_node,`

you would use the following command:

```
# ln -s /tmp/SUNWut/units/IEEE802.mac-address/dev/printers/device_node /dev/usb/sunray-printer
```

Use this soft link (`/dev/usb/sunray-printer`) as the Device URI while creating the print queue.

9. Update `/etc/cups/cupsd.conf` to set the `RunAsUser` property to `No`.

10. Restart the `cups` daemon.

```
# /etc/init.d/cups restart
```

15.4.6.3. How to Set Up an Attached Non-PostScript Printer

Printers that do not use PostScript, such as engineering plotters, are best supported by third-party software. Low-cost inkjet printers require third-party software such as the following:

- Easy Software's ESP PrintPro, available from <http://www.easysw.com>
- Ghostscript, available from <http://www.ghostscript.com>
- Vividata PShop, available from <http://www.vividata.com>

Check with the vendors for pricing and the precise printer models supported.

15.5. Accessing USB Mass Storage Devices

This section provides information about how to manage USB mass storage devices connected to a Sun Ray Client.

15.5.1. Device Nodes and Links (Oracle Solaris)

Mass storage devices have two types of device nodes, block and raw, which are created in the client's `dev` directory. A link to the block device is created in the client's `dev/dsk` directory and a link to the raw device is created in the `dev/rdisk` directory.

Device links have a suffix denoting their slice number. Slice `s2` is known as the backup slice, signifying the complete disk. Other slices are numbered accordingly on the file system on the disk. For UFS disks, slice numbers are derived from the disk label. For FAT disks, slices (partitions in this case) are numbered starting from `s0`. Disk operations such as format or eject should be directed at slice `s2`. Partition operations such as `mount` or `fstyp` should be directed at the individual slice concerned. See [Section 15.5.5, "Mass Storage Devices and Idle Sessions"](#) for examples.

15.5.2. Device Nodes and Links (Oracle Linux)

Mass storage device nodes are block special nodes. They are created in the `dev/dsk` directory. Note that for mass storage devices, device nodes are not created in the devices directory and no device links are created.

Device nodes are named with a partition identifier suffix. The device node representing the whole disk does not have such a suffix. For example:

- `disk3p2` represents partition 2 of disk3.
- `disk3` represents the whole disk.

Disk operations such as eject should be directed at the whole disk. Partition operations such as mount should be directed at individual partitions. See [Table 15.2, "Commands for Common Disk Operation \(Oracle Linux\)"](#) for examples.

15.5.3. Mount Points

When a mass storage device is plugged into the client, if it has an OS-recognizable file system, it is automatically mounted on a directory under the user's mount parent directory. The mount parent directory

is located in `$DTDEVROOT/mnt/`. The user can also locate mount points by using the `-l` option of the `utdiskadm` command.

```
% utdiskadm -l
```

15.5.4. Device Ownership and Hotdesking

When the user's session disconnects from the client, the user loses access rights to the mass storage device, and all pending I/O to the device halts. This situation can cause the data on the device to be corrupted. Users should use `utdiskadm -r` to unmount all file systems safely before hotdesking or unplugging the disk from the client. They should also close all references to files and directories in the mount point to ensure that the device in question is not busy.

15.5.5. Mass Storage Devices and Idle Sessions

If you are using Remote Hotdesk Authentication (RHA), Non-Smart Card Mobility (NSCM), or smart card-based authentication, long I/O operations might fail when using mass storage devices on Sun Ray Clients.

If these types of sessions become idle due to keyboard and mouse inactivity long enough to activate the screen lock, the session is detached. The user loses access to the storage device, causing any I/O in progress to halt, and data may become corrupted.

To avoid this situation, the following options are available:

- Maintain keyboard or mouse activity
- Increase the screen lock idle time sufficiently to allow I/O operations to complete
- Disable the screen lock program
- Disable the NSCM or RHA policies
- Find an alternative way to perform the I/O operation more securely, for example, plug the device directly into the Sun Ray server in a locked server room



Note

Some of these options have security and convenience implications that should be carefully weighed against the timeout issue to determine what is best for your site.

15.5.6. Commands for Common Disk Operation (Oracle Solaris)

Table 15.1, “Commands for Common Disk Operation (Oracle Solaris)” is a summary of common disk operations and the commands used to perform them. Refer to the Oracle Solaris documentation and man pages for more information on the individual commands.

Table 15.1. Commands for Common Disk Operation (Oracle Solaris)

Operation	Command	Device Name Argument Examples (SPARC)	Device Name Argument Examples (x86)
Format	<code>rmformat</code>	Path of whole disk <code>\$UTDEVROOT/dev/rdisk/disk3s2</code>	Path of whole disk <code>\$UTDEVROOT/dev/rdisk/disk3p0</code>
Create file system	<code>mkfs</code>	Path of partition <code>\$UTDEVROOT/dev/rdisk/disk3s0</code>	Path of partition <code>\$UTDEVROOT/dev/rdisk/disk3p1</code>
Create UFS file system	<code>newfs</code>	Path of slice <code>\$UTDEVROOT/dev/rdisk/disk3s0</code>	Path of slice <code>\$UTDEVROOT/dev/rdisk/disk3s0</code>

Operation	Command	Device Name Argument Examples (SPARC)	Device Name Argument Examples (x86)
Mount	<code>utdiskadm -m</code>	Partition name <code>disk3s0</code>	Partition name <code>disk3p1</code>
Unmount	<code>utdiskadm -u</code>	Mount point <code>\$DTDEVROOT/mnt/label1</code>	Mount point <code>\$DTDEVROOT/mnt/label1</code>
Prepare to unplug	<code>utdiskadm -r</code>	Device alias <code>disk3</code>	Device alias <code>disk3</code>
Eject media	<code>utdiskadm -e</code>	Device alias <code>disk3</code>	Device alias <code>disk3</code>
Check for media	<code>utdiskadm -c</code>	Device alias <code>disk3</code>	Device alias <code>disk3</code>
Create <code>fdisk</code> table	<code>fdisk</code>	Path of whole disk <code>\$UTDEVROOT/dev/rdisk/disk3s2</code>	Path of whole disk <code>\$UTDEVROOT/dev/rdisk/disk3p0</code>
Repair file system	<code>fsck</code>	Path of raw slice <code>\$UTDEVROOT/dev/rdisk/disk3s0</code>	Path of raw partition <code>\$UTDEVROOT/dev/rdisk/disk3p1</code>
Display file system capacity	<code>df -k</code>	Mount point <code>\$DTDEVROOT/mnt/label1</code>	Mount point <code>\$DTDEVROOT/mnt/label1</code>
Display slice capacity	<code>prtvtoc</code>	Path of backup slice <code>\$UTDEVROOT/dev/rdisk/disk3s2</code>	Path of backup slice <code>\$UTDEVROOT/dev/rdisk/disk3s2</code>
List devices	<code>utdiskadm -l</code>	None	None

15.5.7. Commands for Common Disk Operation (Oracle Linux)

Table 15.2, “Commands for Common Disk Operation (Oracle Linux)” is a summary of common disk operations and the commands used to perform them.

Table 15.2. Commands for Common Disk Operation (Oracle Linux)

Operation	Command	Device Name Argument Examples
Create file system	<code>mkfs</code>	Path of partition <code>\$UTDEVROOT/dev/dsk/disk3p1</code>
Mount	<code>utdiskadm -m</code>	Partition name <code>disk3p1</code>
Unmount	<code>utdiskadm -u</code>	Mount point <code>\$DTDEVROOT/mnt/label1</code>
Prepare to unplug	<code>utdiskadm -r</code>	Device alias <code>disk3</code>
Eject media	<code>utdiskadm -e</code>	Device alias <code>disk3</code>
Check for media	<code>utdiskadm -c</code>	Device alias <code>disk3</code>
Create <code>fdisk</code> table	<code>fdisk</code>	Path of whole disk <code>\$UTDEVROOT/dev/dsk/disk3</code>
Repair file system	<code>fsck</code>	Path of partition <code>\$UTDEVROOT/dev/dsk/disk3p1</code>
Display file system capacity	<code>df -k</code>	Mount point <code>\$DTDEVROOT/mnt/label1</code>
List devices	<code>utdiskadm -l</code>	None

15.5.8. How to Unmount a Mass Storage Device From a Client



Note

Oracle Linux does not immediately write data to disks. Failure to run `utdiskadm -r` before unplugging mass storage devices will cause loss of data and stale mount points. Make sure to run `utdiskadm -r` before unplugging any mass storage device.

```
% /opt/SUNWut/bin/utdiskadm -r device_name
```

15.5.9. Troubleshooting Mass Storage Devices

This section provides troubleshooting information for mass storage.

15.5.9.1. Problem: Device nodes are not created.

Check the log file `/var/opt/SUNWut/log/utstoraged.log` for a message about why device nodes were not created. Some mass storage device types are not supported.

15.5.9.2. Problem: The device is not automatically mounted.

Check the log file `/var/opt/SUNWut/log/utmountd.log` for an error message.

This condition occurs when the Sun Ray operating system does not recognize the storage devices's file system.

15.5.9.3. Problem: The device is not automatically unmounted.

This condition occurs when a user still has an open reference to the mount point at the time the storage device is unplugged or the user's session is disconnected. The mount point becomes a stale mount point and persists until the system is rebooted or until the administrator removes it.

Use the following procedure to find and remove stale mount points.

1. Search for stale mount points:

```
# utdiskadm -s
```

2. For each stale mount point, close all references to the mount point.
3. For each stale mount point, terminate all processes that refer to the mount point.
4. Remove the mount point.

```
# umount stale_mount_path
```

15.6. USB Headsets

There are a number of USB headsets that have been tested to work on Sun Ray 2 Series and Sun Ray 3 Series Clients.

Once the latest firmware is installed on a Sun Ray Client, no other special installation is required for USB headsets to work. When using USB headsets in a Windows session ([uttsc](#)), the USB redirection feature is not required. However, for Windows XP and Windows Server 2003 R2, you do need to install the Audio Input component. See [Section 17.4, "Audio Input"](#) for details.

For some USB headsets, you must use the `utsettings` command ([Sun Ray Settings GUI](#)) to change the mute and volume settings, even though there may be inline buttons or controls on the headset. For most of the USB headsets, the Sun Ray Settings GUI can be used to adjust the audio settings.

See [Section 13.2.3, "How to Change Sun Ray Client Audio and Display Settings \(Sun Ray Settings GUI\)"](#) for details.

15.6.1. Tested USB Headsets

For the list of tested USB Headsets, see the [Sun Ray Client and Oracle Virtual Desktop Client Peripherals](#) document.

**Note**

When using a USB headset in a Window session (`uttsc` command or fullscreen Windows kiosk mode), the Sun Ray Settings GUI is not available. In this situation, use the audio settings from the Windows desktop.

**Note**

When using the Sun Ray Settings GUI, setting Mic Gain to 0 or de-selecting the Microphone button will not mute the microphone. To ensure the microphone is muted, disable the audio input in your application.

When using `uttsc`, sound input redirection from the Sun Ray Client to the Windows server is disabled by default, which means the microphone is muted for all applications running on the Windows server. You need to use the `uttsc -r soundin:` option to enable it.

15.6.2. Tested Applications

The following applications have been tested to work with USB headsets when using a remote desktop session through the Windows connector:

- Skype 5.2 - Windows 7, Windows XP
- Ekiga - Windows 7
- Adobe ConnectNow - Windows 7

15.6.3. Additional Notes

- When both an analog headset and a USB headset are connected to a Sun Ray Client, the USB headset is used.
- Only one USB headset per Sun Ray Client is supported. If more than one USB headset is connected to a Sun Ray Client at the same time, the last USB headset connected will work. Disconnecting any one of the USB headsets will cause the audio to switch back to the Sun Ray Client's built-in audio device.
- Connecting two different USB audio devices to a Sun Ray Client is not supported. For example, connecting one USB speaker and one USB microphone to a Sun Ray Client is not supported.
- A USB headset and a Sun Ray Client's built-in speaker cannot provide audio output at the same time.
- When using the `uttsc` command, you can use the `-r sound:` and `-r soundin:` options of the `uttsc` command to change the sample rate quality of the audio input or output for a Windows session.

15.7. USB Device Operations Failing After Idle Timeout Limit

If a user fails to access a given session for longer than the screen lock idle timeout interval while an application is accessing a USB device -- for instance, while copying a large number of files to or from a USB flash drive -- the session will be locked. With RHA, NSCM, and authenticated smart cards, this means the session detaches and all USB devices disconnect from the session. This can interrupt or abort the application's access to the device.

The following workarounds are available:

- Advise users to monitor their USB device usage to avoid being timed out

- Set the timeout interval value high enough to allow I/O to complete before the interval elapses
- Disable the screen saver
- Disable RHA

**Note**

The last two alternatives are less desirable because they each remove a level of security. Also, screen savers and RHA are not applicable to kiosk mode.

Chapter 16. Troubleshooting Icons

Table of Contents

16.1. On-Screen Display (OSD) Icons	211
16.2. Server Policy Icons	212
16.3. Troubleshooting Icon Quick Reference	213
16.4. DHCP State Codes	215
16.5. Encryption and Authentication States	216
16.6. Power LEDs	216
16.7. (1) Sun Ray Client Startup Icon	217
16.8. (2) Firmware Download in Progress Icon	217
16.9. (3) Updating Firmware Icon	218
16.10. (4) Firmware Download Diagnostics Icon	218
16.11. (11-14) Network Status Icons	219
16.12. (15) Session Refused Icon	220
16.13. (16) Bus Busy Icon	220
16.14. (20) 802.1x Authentication Icon	221
16.15. (21) Network Connection Verified Icon	221
16.16. (22) Waiting to Connect to Authentication Manager Icon	222
16.17. (23) No Ethernet Signal Icon	223
16.18. (25) Redirection Icon	224
16.19. (26) Wait for Session Icon	224
16.20. (27) DHCP Broadcast Failure Icon	225
16.21. (28) Establishing VPN Connection Icon	225
16.22. (29) VPN Connection Established Icon	226
16.23. (30) VPN Connection Error	226
16.24. (31-34) Network Status Icons	226
16.25. (41-44) Network Status Icons	227
16.26. (46) No Access to Server Icon	228
16.27. (47) No Access for Oracle Virtual Desktop Clients Icon	228
16.28. (48) No Access: Registration Required Icon	228
16.29. (49) No Access: Key Rejected Icon	229
16.30. (50) No Access: Security Policy Violation Icon	229
16.31. (51-54) Network Status Icons	229
16.32. (60) Insert Card Icon	229
16.33. (61) Waiting for Primary Sun Ray Client Icon	230
16.34. (62) Token Reader Icon	230
16.35. (63) Card Error Icon	230
16.36. (64) Waiting For Access Icon	231

This chapter provides details on the troubleshooting icons that display when a Sun Ray Client boots. Some of the server policy icons also display for an Oracle Virtual Desktop Client.

16.1. On-Screen Display (OSD) Icons

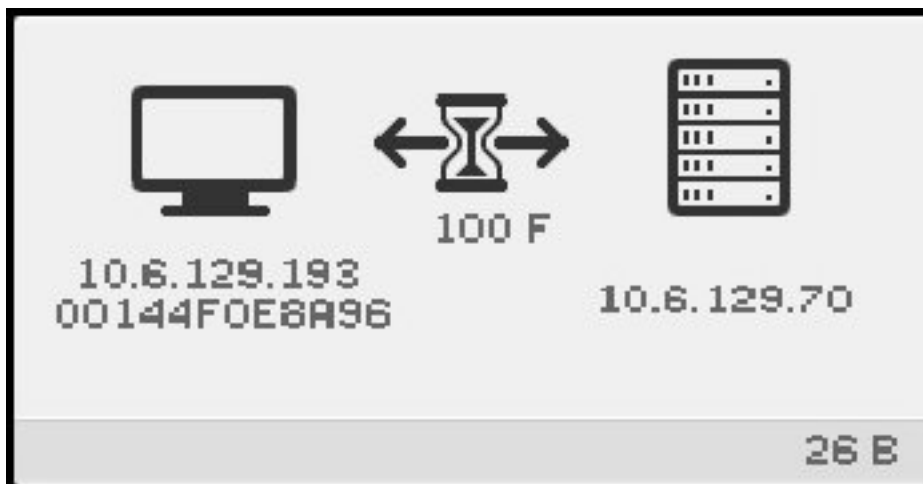
When a Sun Ray Client boots, a spinning wheel icon is displayed by default. To show the details of the boot process and to help identify problems, you can enable the on-screen display (OSD) icons by pressing Stop-O (or Ctrl-Pause-O on non-Oracle keyboards). This condition persists across reboots, so if you want to disable the OSD icons, you need to press Stop-O again.

OSD icons are also displayed if you simultaneously press the three audio keys, if a warning or error occurs on the Sun Ray Client while it boots, or if a state change does not occur for more than ten seconds.

If you see older versions of the icons, the firmware has not been upgraded or is failing. To make sure that the Sun Ray Client is using the latest firmware, see [Chapter 14, Sun Ray Client Firmware](#).

OSD icons are shown as rectangular, light grey icons, such as the following example:

Figure 16.1. On-Screen Display Icon Example



OSD icons can display even if the client is not connected to a server, and they typically provide the following detailed information:

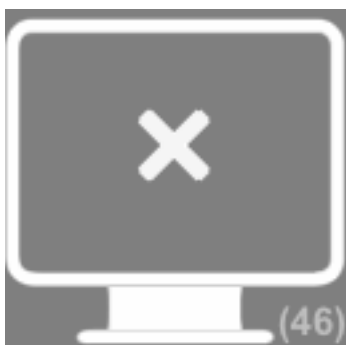
- A unique graphic
- Ethernet address
- IP address of the Sun Ray Client
- Status of link to Sun Ray server
- IP address of authentication server
- Numeric code for [icon message](#)
- Alphabetic code for [DHCP State](#)
- Encryption and authentication information, when appropriate
- Alphabetic code for [Firmware Download Error Codes](#)

16.2. Server Policy Icons

Server policy icons indicate a problem based on a specific server policy that needs attention. They are sent by the server in place of a regular session, they may be overlaid by a concurrent client state OSD icon, and they are not available if the client is behind a NAT router.

Server policy icons are shown as dark grey icons, such as the following example:

Figure 16.2. Server Policy Icon Example



16.3. Troubleshooting Icon Quick Reference



Note

Pressing the Mute-Softer-Louder keys or Ctrl-Pause-N keys simultaneously display the Network Status icons 31-34 or 51-54.

Table 16.1. Troubleshooting Icon Quick Reference

Icon Code (Click for More Information)	General Category	Meaning
1	Startup	Sun Ray Client is starting up and is waiting for Ethernet link.
2	Firmware Download	Sun Ray Client is downloading new firmware.
3	Firmware Download	Sun Ray Client is updating its firmware.
4	Firmware Download	Either the download or update of new firmware has failed.
5	Session Connection	There is no session to connect with the Sun Ray.
6	Session Connection	The server is denying access to the Sun Ray.
7	Smart Card	Local PIN entry to the smart card has failed.
8	Smart Card	In local smart card PIN entry mode.
9	USB	There is an "overcurrent" condition on the USB bus, that is, the total number of devices draws too much current. Consider using a powered hub.
11	Network Status	The network link is up, the server is authenticated and the graphic/keyboard network connection is encrypted.
12	Network Status	The network link is up, the server is not authenticated and the graphic/keyboard network connection is encrypted.
13	Network Status	The network link is up, the server is authenticated and the graphic/keyboard network connection is not encrypted.
14	Network Status	The network link is up, the server is not authenticated and the graphic/keyboard network connection is not encrypted.
15	Session Connection	Sun Ray Client is refusing to talk to the server due to the server's refusal or inability to authenticate or encrypt the network connection.
16	USB	USB bus is busy servicing a high-speed device, and the keyboard or mouse might not be responsive to user input.

Icon Code (Click for More Information)	General Category	Meaning
20	Authentication	Sun Ray Client is attempting 802.1x authentication and this icon indicates the current progress.
21	Startup	Sun Ray Client is booting up and is waiting for DHCP IP address and parameter assignment.
22	Startup	Sun Ray Client is booting up and is waiting for the initial connection to a Sun Ray server.
23	Network Status	The connection between the Sun Ray Client and the network is down. Check the network drop cable. If the network drop cable is okay, check the network switch.
24	Session Connection	Sun Ray Client has disconnected from the previous server.
25	Session Connection	Sun Ray Client is being redirected to a new server.
26	Session Connection	Sun Ray Client has connected to the server and is waiting for graphics traffic.
27	Startup	Sun Ray Client is broadcasting to locate a Sun Ray server since either it was not provided with Sun Ray specific DHCP parameters or all of the specified servers are not responding.
28	Startup	VPN connection being attempted.
29	Startup	VPN connection established.
30	Startup	VPN connection error.
31	Network Status	The network link is up, the server is authenticated, and graphics/keyboard network connection is not encrypted.
32	Network Status	The network link is up, the server is not authenticated, and graphics/keyboard network connection is encrypted.
33	Network Status	The network link is up, the server is authenticated, and graphics/keyboard network connection is encrypted.
34	Network Status	The network link is up, the server is not authenticated, and graphics/keyboard network connection is not encrypted.
35	Startup	Sun Ray Client has been disconnected from its server, either by a STOP-Q session disconnect event or by the VPN session timeout value having been set and exceeded.
41	Network Status	The network link is up, the server is authenticated, the client is authenticated, and the graphic/keyboard network connection is encrypted.
42	Network Status	The network link is up, the server is not authenticated, the client is authenticated, and the graphic/keyboard network connection is encrypted.
43	Network Status	The network link is up, the server is authenticated, the client is authenticated, and the graphic/keyboard network connection is not encrypted.
44	Network Status	The network link is up, the server is not authenticated, the client is authenticated, and the graphic/keyboard network connection is not encrypted.

Icon Code (Click for More Information)	General Category	Meaning
46	Server Policy	No access to server.
47	Server Policy	No access for Oracle Virtual Desktop Clients.
48	Server Policy	No access: registration required.
49	Server Policy	No access: client key is rejected.
50	Server Policy	No access: security policy violation.
51	Network Status	The network link is up, the server is authenticated, the client is authenticated, and graphics/keyboard network connection is not encrypted.
52	Network Status	The network link is up, the server is not authenticated, the client is authenticated, and graphics/keyboard network connection is encrypted.
53	Network Status	The network link is up, the server is authenticated, the client is authenticated, and graphics/keyboard network connection is encrypted.
54	Network Status	The network link is up, the server is not authenticated, the client is authenticated, and graphics/keyboard network connection is not encrypted.
60	Server Policy	Insert card. If the site's authentication policy allows access only by card, this icon is displayed to prompt the user to insert a card. Access without a card is disabled.
61	Server Policy	Waiting for primary Sun Ray Client. The client is a secondary client in a multihead group, and the primary client is not currently connected.
62	Server Policy	Token reader. The Sun Ray Client is a token reader. When a site policy disallows pseudo-sessions, a Sun Ray Client configured as a token reader displays the Token Reader icon instead of the Login dialog box.
63	Server Policy	Smart card not recognized. The smart card is not recognized by the Sun Ray server or there is a reader error.
64	Server Policy	Waiting for session access. Access is temporarily denied, but the Sun Ray Client automatically retries when this condition is resolved.

16.4. DHCP State Codes

Some icons also show a state code after the number to provide more information, which are described in [Table 16.2, "DHCP State Codes"](#).





Table 16.2. DHCP State Codes

DHCP State Code	Meaning
A	DHCP only provided IP address with no additional parameters.
B	DHCP provided IP address, subnet mask, and router, but Sun Ray vendor-specific parameters are missing.
C	DHCP provided IP address and Sun Ray vendor-specific parameters, but subnet mask and router are missing.
D	DHCP provided all expected parameters.

16.5. Encryption and Authentication States

In many of the OSD icons, symbols are used to indicate the encryption and authentication states for the Sun Ray Client, which are described in [Table 16.3, “Encryption and Authentication States”](#). These symbols are typically seen in the Network Status icons.

Table 16.3. Encryption and Authentication States

Symbol	Meaning
	Graphics/keyboard network connection is encrypted.
	Graphics/keyboard network connection is not encrypted.
	Server/client is authenticated.
	Server/client is not authenticated.

16.6. Power LEDs

[Table 16.4, “Power LEDs”](#) lists the actions to take based on the state of the power LED on Sun Ray Clients.

Table 16.4. Power LEDs

Sun Ray Client Hardware State	Action to Take
Off	Check to see whether the Sun Ray Client is plugged in. Replace the Sun Ray Client.
Green	Normal operation.
Amber	Hardware fault. Replace the Sun Ray Client.
Blinking	PROM is corrupted. Check that firmware downloads are configured and enabled, then power cycle the Sun Ray Client.
Card reader LED remains on even when card is removed	Card reader hardware problem. Replace the Sun Ray Client.

16.7. (1) Sun Ray Client Startup Icon

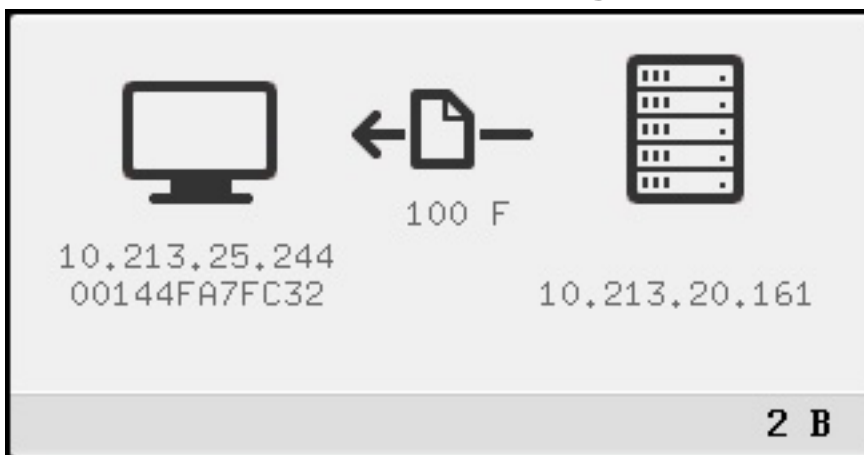


The Sun Ray Client Startup icon indicates that the Sun Ray Client has passed the power-on self test but has not yet detected an Ethernet signal. The icon is displayed for a few seconds as part of the normal startup process. When an Ethernet signal is detected, the Network Connection Verified (21) OSD is displayed.

Problem: The Sun Ray Client Startup OSD is displayed for more than 10 seconds.

- Check that the Ethernet cable is plugged into the Sun Ray Client correctly and that the other end is plugged in to the correct hub, switch, or network outlet.
- If the Sun Ray Client is connected through a hub or a switch, verify that the hub or switch is powered on and configured correctly. A link LED on the switch or hub indicates that the connection is alive.

16.8. (2) Firmware Download in Progress Icon



The Firmware Download in Progress icon indicates that the Sun Ray Client is downloading new firmware from a Sun Ray server. The following message is also displayed: [Reading firmware](#).

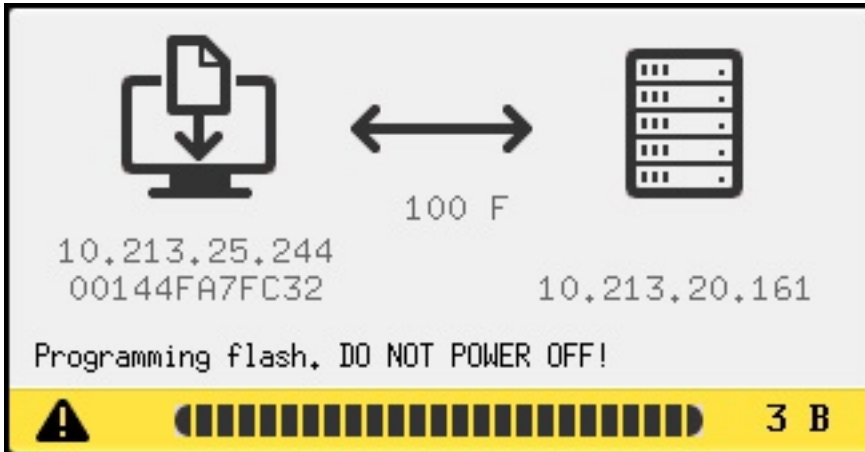
Downloading the firmware takes less than a minute. If you interrupt the download, the Sun Ray Client has to download the new firmware the next time it reboots.



Note

When downloading the firmware on Sun Ray 2 Series Clients, pixel artifacts fill the screen. This is normal.

16.9. (3) Updating Firmware Icon



The Updating Firmware icon indicates that the downloaded firmware is being updated on the Sun Ray Client. The following messages are also displayed:

- Erasing flash. DO NOT POWER OFF!
- Programming flash. DO NOT POWER OFF!

The Sun Ray Client reboots after updating the firmware.



Caution

Do not interrupt the firmware update. Interrupting the firmware update may make the Sun Ray Client unusable.



Note

If there is an additional firmware update for the internal smart card controller, the Updating Firmware icon is displayed again and the smart card LED flashes while the update occurs. The following message is also displayed: [Programming Smart Card Controller. DO NOT POWER OFF!](#). The Sun Ray Client reboots after updating the firmware for the internal smart card controller.

16.10. (4) Firmware Download Diagnostics Icon

The Firmware Download icon is displayed with a code or a message when an error occurs during a download of firmware. The following table lists the error codes. These error messages appear in English even in localized versions of Sun Ray Software.



Table 16.5, “Firmware Download Error Codes and Messages” lists the firmware download error codes and messages. Firmware download error codes are valid only with OSD icon 4.

Table 16.5. Firmware Download Error Codes and Messages

Error Code	Error Message
E	FW Load: No server
F	FW Load: Name too long
G	FW Load: Bad read
H	FW Load: Bad signature
I	FW Load: Failed decompression
J	FW Load: Invalid module type
K	FW Load: Version mismatch
L	FW Load: Not enough memory
M	FW Load: Prevented by barrier
N	FW Load: Invalid HW version
O	FW Load: Flash write error

The Firmware Download with a 4 error code icon is displayed when the Sun Ray Client fails to download new firmware. The message "FW Load: Prevented by barrier" indicates that the Sun Ray Client already has a later version of the firmware.

In the `syslog`, the following message indicates that a barrier level has been set to prevent Sun Ray Clients from downloading an earlier version of the firmware.

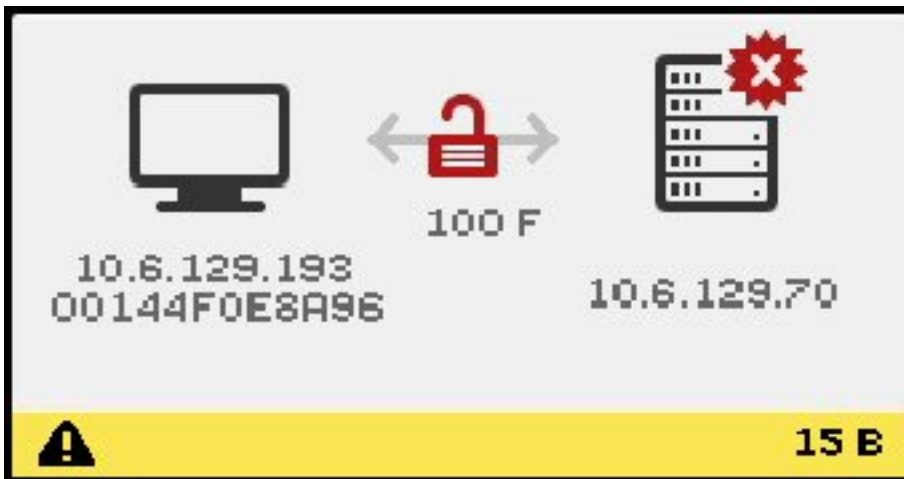
```
Firmware upgrade/downgrade not allowed! Barrier is 310 Firmware level is 0
```

Check `/var/opt/SUNWut/log/messages` to confirm that your configuration is set up properly.

16.11. (11-14) Network Status Icons

The 11-14 Network Status icons are similar to the 31-34 icons. The only difference is the level of the current authentication. See Table 16.1, “Troubleshooting Icon Quick Reference” for more information for a description of each icon.

16.12. (15) Session Refused Icon



The Session Refused icons are displayed during a possible security breach because authentication has failed.

Problem: Icon shows the 15D message.

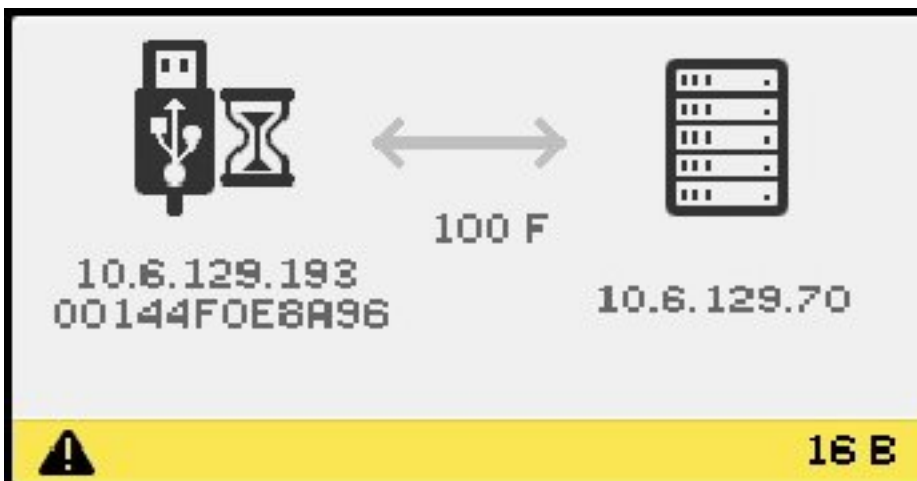
The Sun Ray Client is refusing to connect to a server because it is unable to verify the validity of the Sun Ray server. This error can occur only if an unknown Sun Ray server tries to emulate a valid Sun Ray server. This situation is a session security breach.

Problem: Icon shows the 50D message.

The Sun Ray server is refusing to grant a session to the Sun Ray Client because the client is unable to fulfill the server's security requirements.

- Upgrade the Sun Ray Client's firmware version. This error can occur with firmware versions earlier than 2.0 when the server is configured for hard security mode.
- As an alternative, determine whether your site requires hard security mode. If not, the session can be enabled with soft security mode.

16.13. (16) Bus Busy Icon



The Bus Busy OSD indicates that the Sun Ray USB bus is servicing a high-speed device and the keyboard or mouse might not be responsive to user input.

This icon is displayed during an unusually long print job and disappears when the job is done. No action is needed unless killing the print job is necessary.

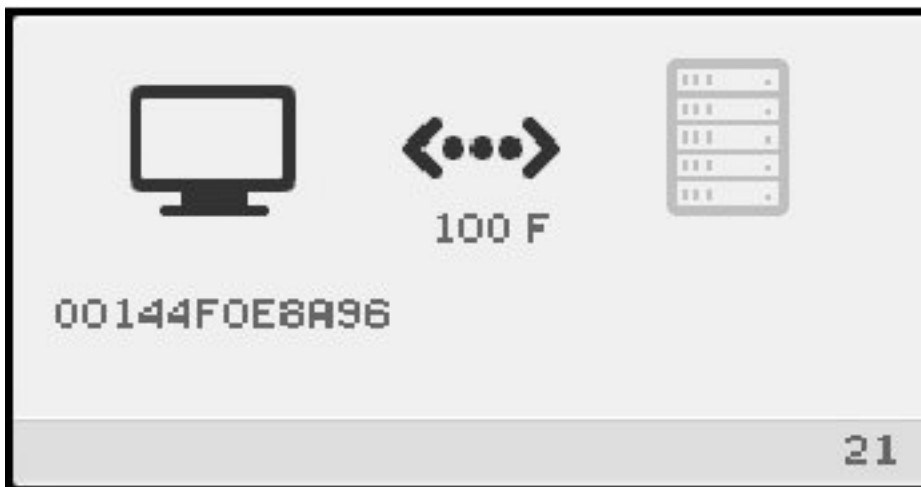
16.14. (20) 802.1x Authentication Icon

Table 16.6, “802.1x Authentication Error Codes and Messages” lists the 802.1x authentication error codes and messages. During the authentication, you may be prompted to provide required `wpa_supplicant` values that are not defined, such as passwords. Also, `wpa_authentication` status messages are displayed as the authentication progresses.

Table 16.6. 802.1x Authentication Error Codes and Messages

Error Code	Error Message
A - D	Authentication has completed
E	802.1x initialized
F	Extended Authentication Protocol (EAP) started
G	EAP succeeded
H	EAP failed

16.15. (21) Network Connection Verified Icon



The Network Connection Verified icon indicates that the Sun Ray Client has detected the Ethernet carrier but has not yet received its initial parameters or IP address from the DHCP server. The icon is displayed for a few seconds as part of the normal startup process.

After the DHCP server has allocated an IP address, the icon is updated with the Sun Ray Client's assigned IP address. When the network connection is verified, the Sun Ray Client connects to the Sun Ray server.

Problem: The icon is displayed for more than 10 seconds.

- Verify that the DHCP server is running and has not run out of IP addresses to assign to clients.
- Verify that the DHCP server is configured properly for network parameters.

Problem: The icon displays an IP address and an icon message, either 21A or 21B, depending on whether the Sun Ray server is on a LAN network or a dedicated interconnect.

This condition occurs when the Sun Ray Client receives an IP address from the DHCP but no other parameters. The Sun Ray Client issues a DHCP_INFORM request to obtain the Sun Ray-specific parameters.

- Code 21 A indicates that the Sun Ray Client received an IP address and is waiting for a response to its DHCP `inform` request.
- Code 21 B indicates that the Sun Ray Client received an IP address and IP router and is waiting for a response to its DHCP `inform` request.

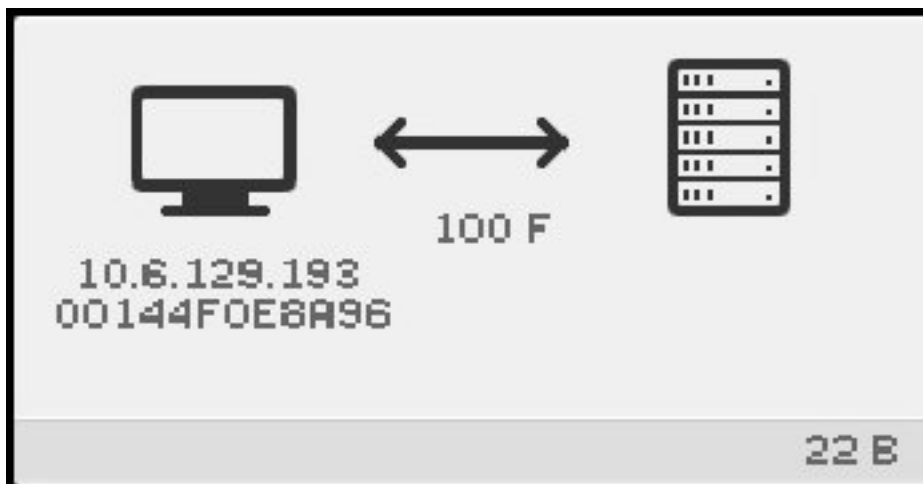
If no response is received, the Sun Ray Client continues the startup process using only the IP address. In a private interconnect or simple LAN configuration, the Sun Ray Client can function successfully. However, performance of the Sun Ray Client might be affected. If the Sun Ray Client is part of a complex LAN configuration, it can fail later in the start up process because it requires the additional parameters and Sun Ray-specific vendor options to handle network operations, such as when a Sun Ray Client is located several hops away from the Sun Ray server's subnet.

Continue with the startup process, if possible, and at the next opportunity, do the following:

- For LAN configurations with other non-Sun Ray DHCP services but no `bootp` proxy agent, verify the DHCP server and the Sun Ray vendor tags.
- For routed configurations, verify that the `bootp` proxy agent is configured correctly in the Sun Ray Client's subnet and that it points to one of the Sun Ray servers in the failover group.
- For non-routed private interconnect configurations, the Sun Ray server performs the functions of a DHCP server. Verify that it is configured properly for DHCP services.

When the Sun Ray Client concludes the interaction with the DHCP server, it connects to a Sun Ray server and then interacts with the server's Authentication Manager, indicated by the Waiting to Connect to Authentication Manager OSD. Occasionally, the Sun Ray Client is first routed to another Sun Ray server. In this case, the Redirection OSD icon is displayed for a few seconds and then, as the Sun Ray Client interacts with the new server's Authentication Manager, the Waiting to Connect to Authentication Manager OSD is displayed.

16.16. (22) Waiting to Connect to Authentication Manager Icon



The Waiting to Connect to Authentication Manager icon indicates that the Sun Ray Client has received its parameters from the DHCP server and it has connected to the Sun Ray server but has not yet completed its authentication. The icon is displayed for a few seconds as part of the normal startup process.

Problem: The icon displays for more than 10 seconds or the Sun Ray Client resets after the icon is displayed.

- Verify that Sun Ray services, including the Authentication Manager, are running on the Sun Ray server.

In a LAN configuration or other routed environment:

- Verify that the Sun Ray Client's IP address can reach the Authentication Manager.
- Verify that the Sun Ray Client's routing information, received from the Sun Ray server, is correct.
- Verify that the `bootp` proxy agent is configured correctly in the Sun Ray Client's subnet and that it points to one of the Sun Ray servers in the failover group.
- Run `utquery` for the Sun Ray Client's IP address to see the parameters that the Sun Ray Client received. If the parameters do not include an `AuthSrvr` parameter, the DHCP server might not have sent the Sun Ray parameters or the parameters might not be correct.
 - To confirm that the DHCP server can be reached, check the value of the `DHCPServer` parameter.
 - To confirm that the DHCP server sends the proper Sun Ray-specific parameter values, check the value of the `INFORMServer` parameter.

If a value is incorrect, look at your `bootp` relay configurations and DHCP server configurations for network and Sun Ray parameters. For details of these parameters, see the `utquery` man page.

- To restart DHCP on a Oracle Solaris server, type the following as superuser:

```
# /etc/init.d/dhcp stop
# /etc/init.d/dhcp start
```

16.17. (23) No Ethernet Signal Icon

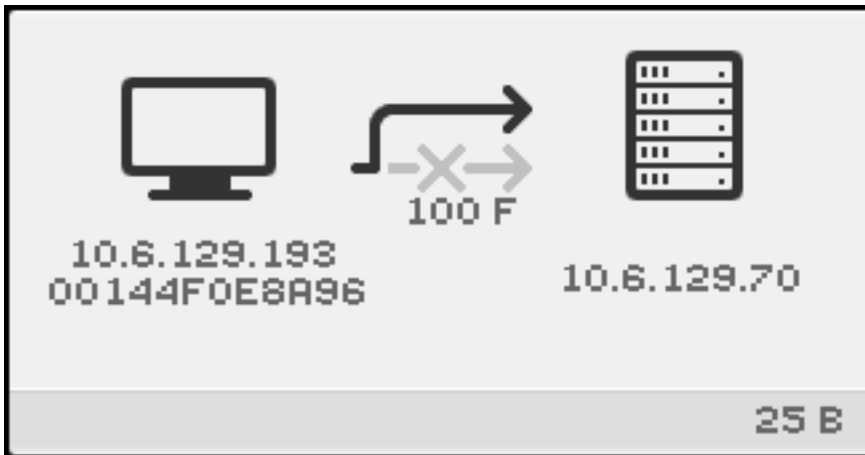


The No Ethernet Signal OSD indicates that the Sun Ray Client had an Ethernet address and an IP address but later lost the Ethernet signal.

- Check that the Ethernet cable has not become unplugged from the Sun Ray Client or from the switch or network outlet.

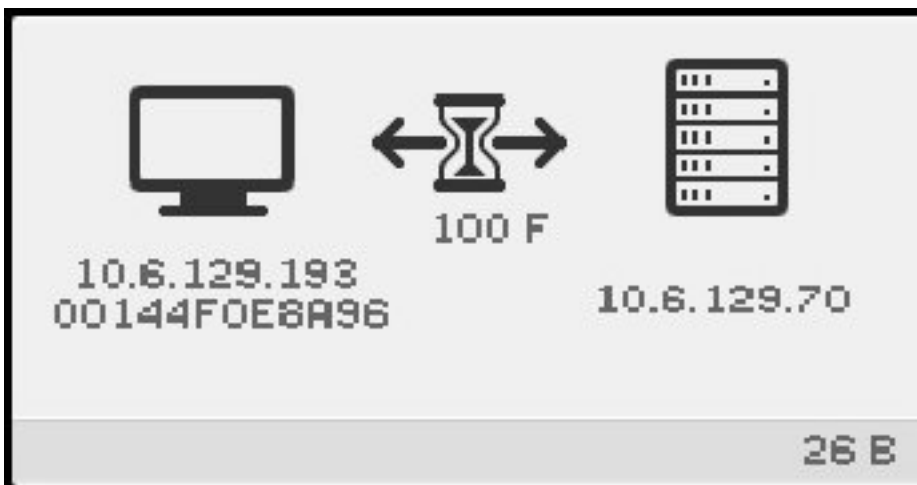
- If the Sun Ray Client is connected through a hub or switch, make sure that the hub or switch is still powered on.

16.18. (25) Redirection Icon



The Redirection OSD indicates that the Sun Ray Client is being redirected to a new Sun Ray server. This redirection can occur for any of several reasons, including load balancing. The icon is displayed for a few seconds while the Sun Ray Client connects to the new Sun Ray server and then the Waiting to Connect to Authentication Manager OSD is displayed.

16.19. (26) Wait for Session Icon



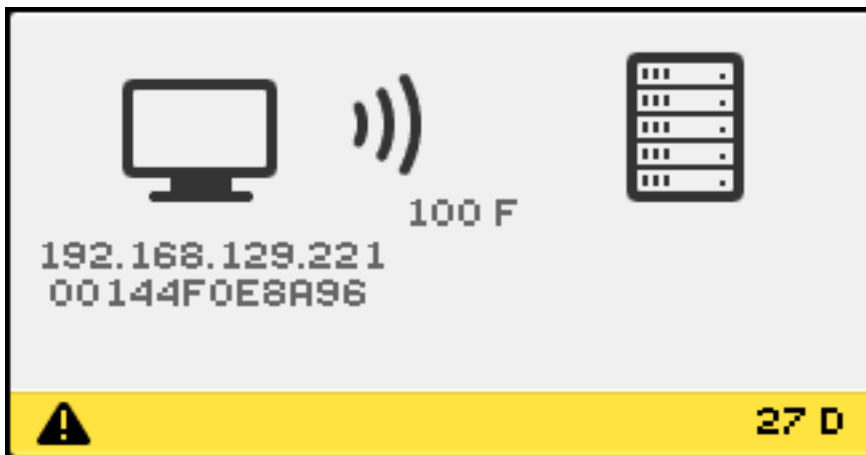
The Wait for Session OSD indicates that the Sun Ray Client is waiting for its X Window session. The icon is displayed for a few seconds as part of the normal startup process.

If this icon is displayed for a long time, display traffic from the server is not arriving to the client. Some possible reasons for this problem are:

- The network (routers, switches, firewalls) is not correctly transmitting UDP traffic from the server to the client.
- The server is attempting to display one of the Server Policy icons, but the client is behind a NAT router or gateway.

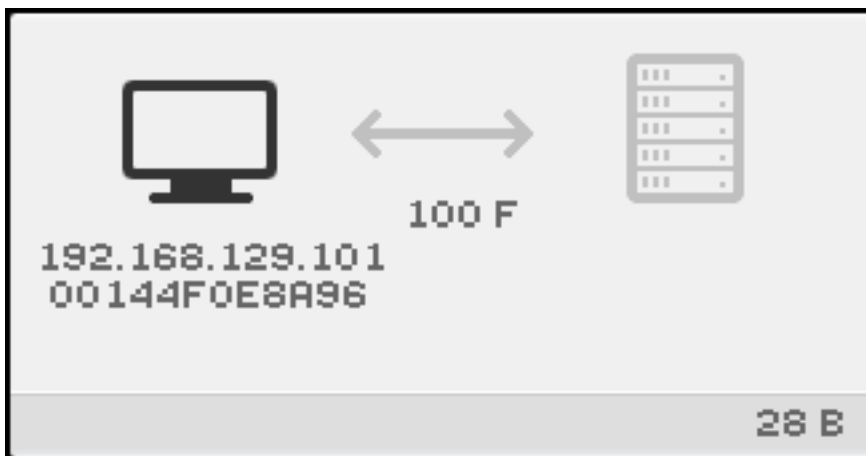
- The X server (Xnewt or Xsun) that is the source of the display traffic on the Sun Ray server side is not working properly. It might be crashed or hung.
- The display manager (`dtlogin` on Oracle Solaris or `gdm` on Oracle Linux) has failed to start an X server for the session. It might be crashed, hung, or not configured properly. If you suspect that `dtlogin` configuration files have been corrupted, see [Section 5.3, “How to Check and Fix Corrupted Configuration Files \(Oracle Solaris 10\)”](#).

16.20. (27) DHCP Broadcast Failure Icon



The DHCP Broadcast Failure icon is displayed if the Sun Ray Client is attempting to locate a Sun Ray server and either no servers respond or the Sun Ray-specific DHCP parameters are not correct.

16.21. (28) Establishing VPN Connection Icon



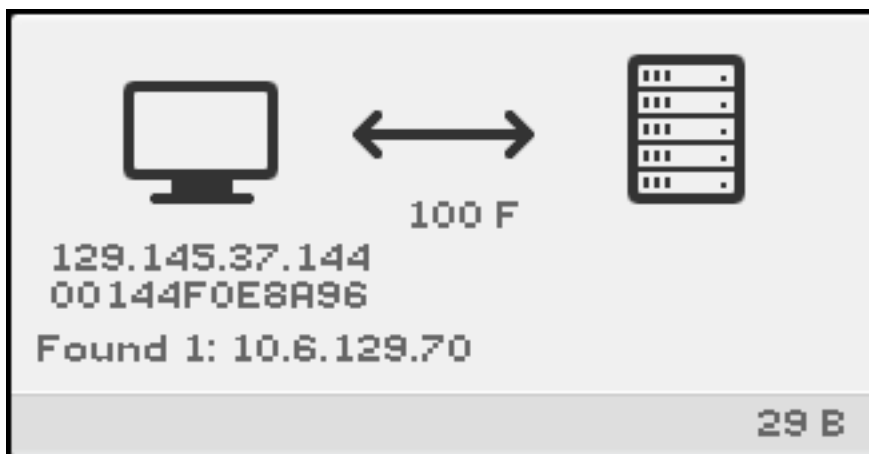
The Establishing VPN Connection icon is displayed while a Sun Ray Client is trying to connect to the Sun Ray server through a VPN connection. This icon can also include one of the following state codes shown in [Table 16.7, “VPN Connection State Codes”](#).

Table 16.7. VPN Connection State Codes

State Code	Meaning
E	VPN Phase 1 IKE initiated.
F	VPN Phase 1 IKE complete.

State Code	Meaning
G	VPN connection expired.
H	VPN Phase 2 initiated.
I	VPN Phase 2 complete.

16.22. (29) VPN Connection Established Icon



When the VPN connection is established, the VPN Connection Established icon is displayed.

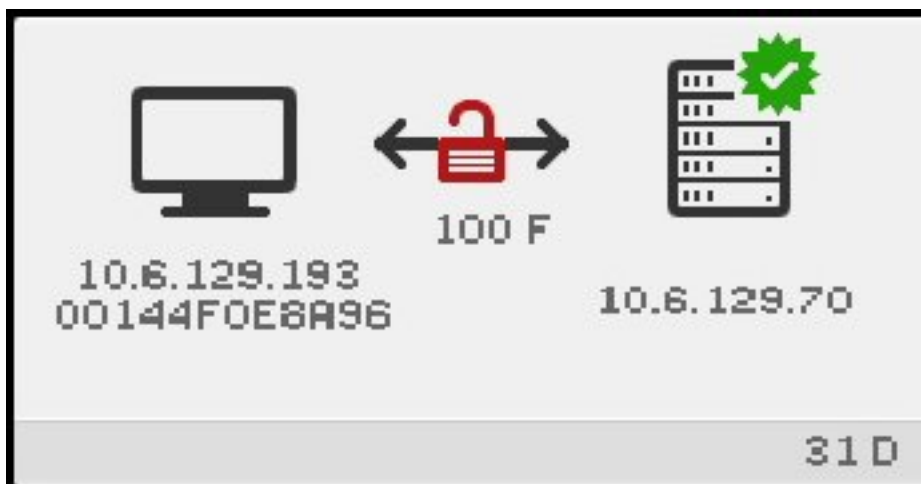
16.23. (30) VPN Connection Error

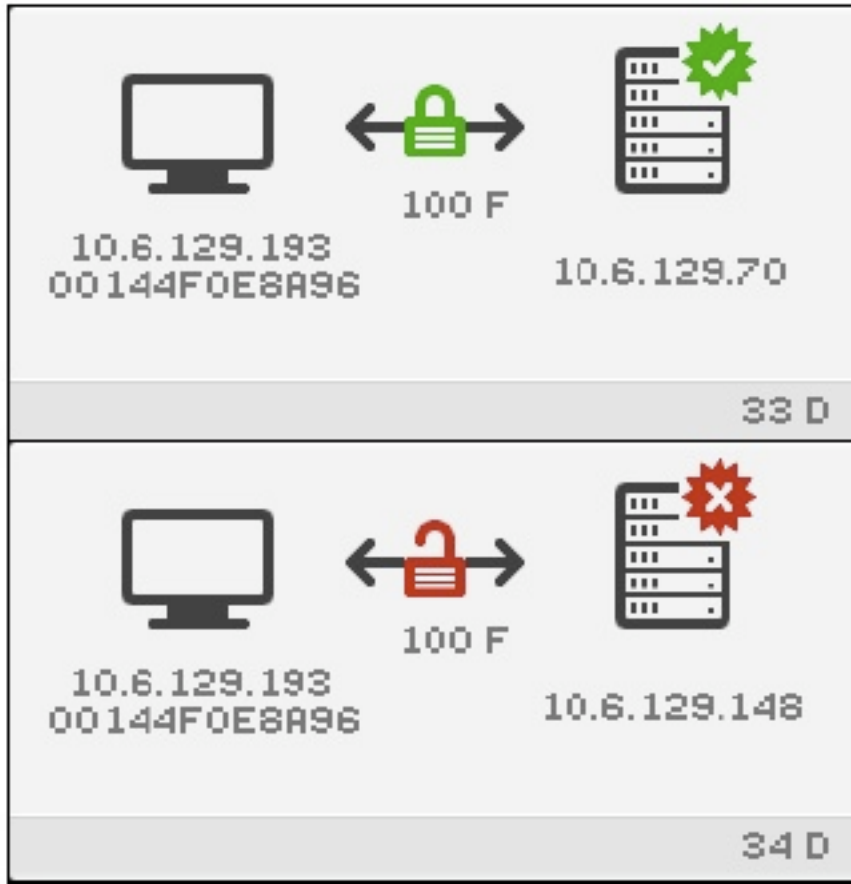
There is a problem with the VPN connection. This icon can also include one of the following state codes shown in [Table 16.7, “VPN Connection State Codes”](#). Make sure the VPN configuration is valid. See [Chapter 14, Sun Ray Client Firmware](#) for more information.

Table 16.8. VPN Connection Error State Codes

State Code	Meaning
L	Invalid group or group key.
K	Invalid Xauth user name or password.

16.24. (31-34) Network Status Icons





The Network Status OSD icon shows the Ethernet address, the assigned IP address, the connected server, encryption status, DHCP state, link speed and link mode.

To display current information about the Ethernet link, do one of the following at any time.

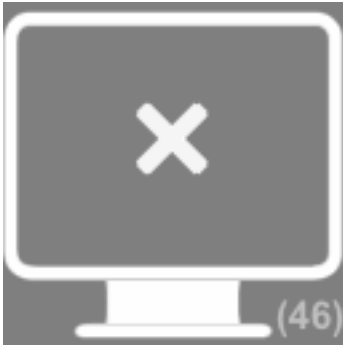
- On an Oracle keyboard, press the three audio volume keys simultaneously. To get the same effect on a non-Oracle keyboard, press Ctrl-Pause-N.
- Disconnect and reconnect the Ethernet cable.

A value of 10 indicates a link speed of 10 Mbps; 100 indicates 100 Mbps. A value of F indicates that the link mode is full duplex. A value of H indicates half-duplex mode.

16.25. (41-44) Network Status Icons

The 41-44 Network Status icons are similar to the 31-34 icons. The only difference is the level of the current authentication. See [Table 16.1, "Troubleshooting Icon Quick Reference"](#) for more information for a description of each icon.

16.26. (46) No Access to Server Icon



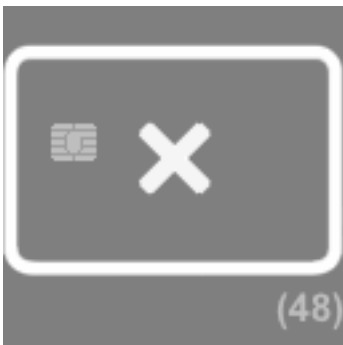
This icon usually displays if the policy disallows card access and a card is inserted.

16.27. (47) No Access for Oracle Virtual Desktop Clients Icon



This icon indicates that access for Oracle Virtual Desktop Clients is disabled. To enable access for Oracle Virtual Desktop Clients, refer to the Oracle Virtual Desktop Client documentation.

16.28. (48) No Access: Registration Required Icon



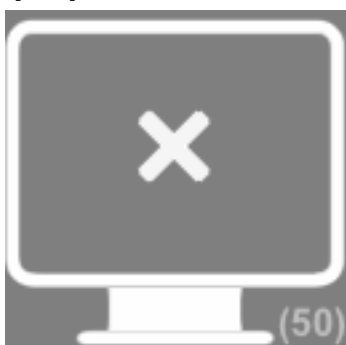
The card or Sun Ray Client is not registered. If ATI is configured for a site, the ATI script is run when this icon is first displayed. If the script registers the card, this state might not last long.

16.29. (49) No Access: Key Rejected Icon



This icon is displayed if only confirmed keys are allowed access by policy. It might be displayed if there is a key conflict, but other icons might display instead.

16.30. (50) No Access: Security Policy Violation Icon

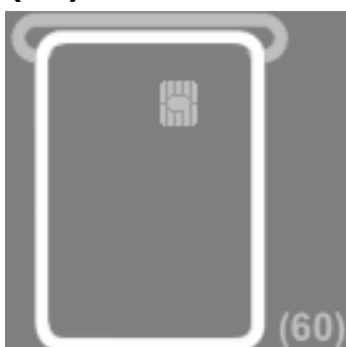


This icon is displayed if the client is running old firmware that does not support encryption or client authentication and the server has "hard" security mode set. This icon might also display in other security-related cases, such as key conflict or failed key validation, but other icons might display instead.

16.31. (51-54) Network Status Icons

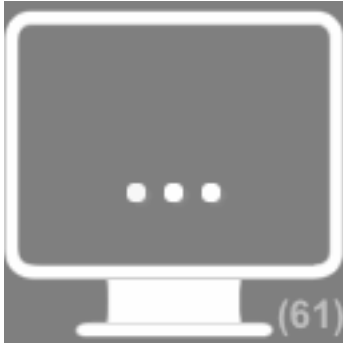
The 51-54 Network Status icons are similar to the 31-34 icons. The only difference is the level of the current authentication. See [Table 16.1, "Troubleshooting Icon Quick Reference"](#) for more information for a description of each icon.

16.32. (60) Insert Card Icon



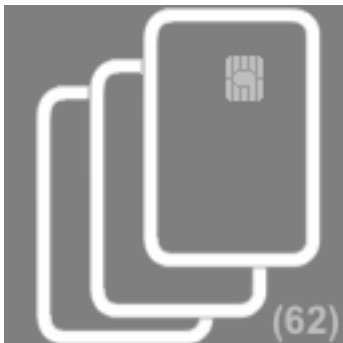
If the site's authentication policy allows access only by card, this icon is displayed to prompt the user to insert a card. Access without card is disabled.

16.33. (61) Waiting for Primary Sun Ray Client Icon



The Sun Ray Client is a secondary client in a multihead group, and the primary client is not currently connected.

16.34. (62) Token Reader Icon



The Sun Ray Client is a token reader. When a site policy disallows pseudo-sessions, a Sun Ray Client configured as a token reader displays the Token Reader icon instead of the Login dialog box.

16.35. (63) Card Error Icon



The smart card is not recognized by the Sun Ray server or there is a reader error. The following reasons may include:

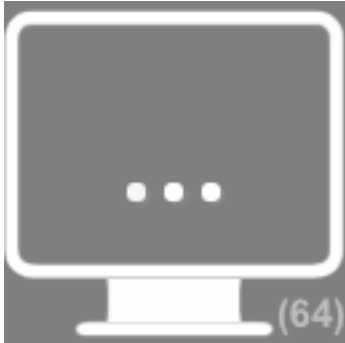
- The Sun Ray Client is running an older firmware.

- The smart card's contacts are dirty, the contacts on the smart card reader are dirty, or the card is not properly inserted.
- The smart card is malfunctioning.
- The Sun Ray server is not configured to read this type of smart card or an error exists in the configuration file.

To fix this problem, try one of the following actions:

- Upgrade the firmware on the Sun Ray Client.
- Clean the smart card.
- Replace the smart card.
- Verify that the Sun Ray server has the appropriate smart card configuration files installed and configured.

16.36. (64) Waiting For Access Icon



This icon indicates that the server is not allowing access at the current time. This problem occurs when a Sun Ray Client loses power or the network connection to the server is interrupted and the smart card from the Sun Ray Client is inserted into a different Sun Ray Client before the server has timed out the lost connection. Because the old connection is still active, new connections using the same smart card are unable to gain access.

When this conditions occurs, the server checks the status of the old connection. After the time reserved for this check has elapsed (an initial default of 10 seconds), the Sun Ray Client connection is restarted and the condition should be automatically resolved. Either the session access is granted or the Sun Ray Client remains in this Waiting for Access state (64).

If a Sun Ray Client continues to remain in this state, the same token is being used with another connection. Specifically, two physical tokens (smart card, Sun Ray Client, Oracle Virtual Desktop Client profile) are trying to connect to the same session.

Possible reasons for this issue include the following:

- A security incident where a copied or fake smart card is used to gain access to the session.
- A security incident where a copy of an Oracle Virtual Desktop Client profile is used to gain access to the session. This situation might also indicate a user error. Oracle Virtual Desktop Client profile files should not be copied to a different computer or user account.
- A registered token policy is in effect, alias tokens have been configured, and an alias token is still connected to the session the user is trying to access. If access is denied because of a currently

connected alias token, the connected alias token needs to be disconnected to regain access. For example, the aliased smart card must be removed from its Sun Ray Client.

Chapter 17. Windows Connector

Table of Contents

17.1. Windows Connector Overview	234
17.2. Requirements	236
17.3. Using the Windows Connector	236
17.3.1. How to Start a Windows Session	236
17.3.2. How to Start a Windows Session Within Java Desktop System (Oracle Solaris 10)	238
17.3.3. How to Lock a Windows Session	238
17.3.4. How to Set Up Access to the <code>uttsc</code> Command	239
17.3.5. How to Set Up a Desktop Shortcut to Start a Windows Session	239
17.3.6. How to Separate Settings for Session Locale and Keyboard Layout	239
17.4. Audio Input	239
17.4.1. Enabling Audio Input on Windows 7 and Windows Server 2008 R2	240
17.5. Video Acceleration	240
17.5.1. Video Acceleration Requirements	240
17.5.2. Videos Accelerated	241
17.5.3. Audio Accelerated	243
17.5.4. Additional Notes	243
17.5.5. How to Enable Video Redirection on Windows Server 2008 R2	245
17.5.6. Video Acceleration Troubleshooting	245
17.6. USB Device Redirection	250
17.6.1. Device Access	251
17.6.2. Supported Configurations	251
17.6.3. Tested USB Devices	251
17.6.4. Additional Notes	251
17.6.5. How to Add USB Drivers to a Virtual Machine	252
17.6.6. USB Redirection Troubleshooting	253
17.7. Hotdesking	254
17.7.1. Hotdesking Behavior	255
17.7.2. Location Awareness	255
17.8. Session Directory	257
17.9. Network Security	257
17.9.1. Built-in RDP Network Security	258
17.9.2. Enhanced Network Security	258
17.10. Automatic Reconnection	260
17.11. Compression	260
17.11.1. How to Disable Compression	260
17.12. Licensing	260
17.12.1. Per-user Mode Versus Per-device Mode	261
17.13. Smart Cards	262
17.13.1. How to Enable Smart Card Readers on a Windows System	262
17.13.2. How to Set Up Smart Card Login for Windows	262
17.14. Multi-Monitor Support	262
17.15. Dynamic Session Resizing	263
17.16. Printing	263
17.16.1. How to Set Up Print Queues (Oracle Solaris 10)	264
17.16.2. How to Set Up Print Queues (Oracle Linux)	264
17.16.3. How to Make Sun Ray Printers Available to a Windows Session	265
17.16.4. How to Manage Printer Configurations for Users	266
17.16.5. How to Set Up Follow-Me Printing	266

17.16.6. Printers Troubleshooting	267
17.17. Accessing Serial Devices	267
17.18. <code>uttsc</code> Error Messages	268
17.18.1. General Troubleshooting	269

This chapter provides information about managing the Windows connector.

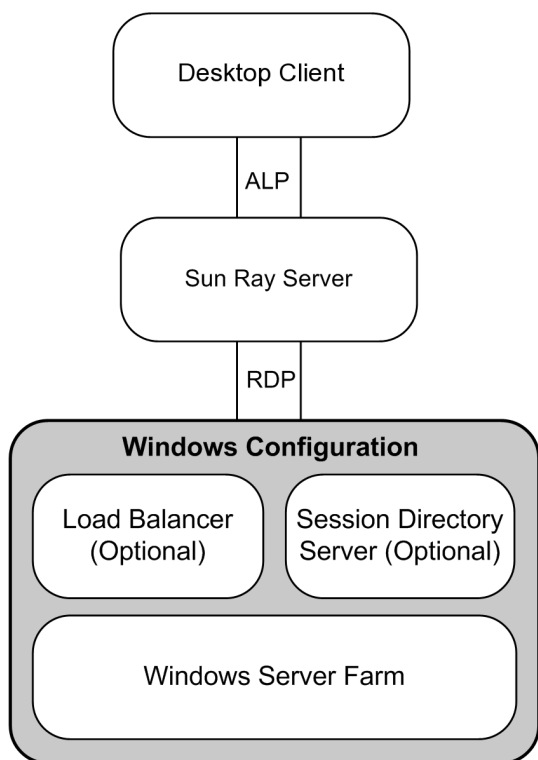
17.1. Windows Connector Overview

The Windows connector is a Microsoft Remote Desktop Protocol (RDP) client that enables Sun Ray users to access applications running on remote Microsoft Windows systems. This client is especially useful to those who are accustomed to Windows-based applications or who want to access documents in certain formats from a desktop client, either a Sun Ray Client or an Oracle Virtual Desktop Client. Users can access their Windows desktop on a desktop client at full screen or in a window.

Kiosk mode is the recommended way to configure the Windows connector. By using kiosk mode, you can set up a desktop client to behave just like a Windows system, which means users do not have to interact with the Oracle Solaris or Oracle Linux login screen and they do not need to specify the `uttsc` command. See [Chapter 10, Kiosk Mode](#) for more information.

As stated, the Windows connector mediates between the desktop client and a Windows system. It uses the Remote Desktop Protocol (RDP) to communicate with the Windows system and the Appliance Link Protocol (ALP) to communicate with the desktop client, as shown in [Figure 17.1, “Windows Connector Overview”](#).

Figure 17.1. Windows Connector Overview



Once the Windows connector is configured, a user has to use the `uttsc` command to connect to a Windows system if the kiosk mode is not configured. The user can modify the command to accommodate a variety of preferences or options, such as specifying screen size or a list of available printers.

Table 17.1, “Windows Connector Features” lists the features provided by the Windows connector.

Table 17.1. Windows Connector Features

Feature	Description
Section 17.6, “USB Device Redirection”	Enables users to access USB devices connected to a Sun Ray Client from their Windows sessions, provided that the appropriate device drivers are installed on the Windows system.
Section 17.5, “Video Acceleration”	The Windows connector provides features to increase the performance for video streams and Adobe Flash content. The support provided is dependent on the Windows OS.
Section 17.4, “Audio Input”	Users can play sound files on their Sun Ray Clients (audio out) with audio applications located on the Windows system. Recording from the Sun Ray Client to the Windows system (audio in) is also supported.
Clipboard	<p>The Windows connector enables cut-and-paste text functionality between Windows applications and the applications running on the Oracle Solaris or Oracle Linux desktop. Copying and pasting is enabled for all supported languages, including double-byte languages such as Chinese, Japanese, and Korean. The Windows connector does not support copying and pasting functionality for Rich Text Format.</p> <p>The following behaviors, although similar, are caused by limitations in different applications:</p> <ul style="list-style-type: none"> • Once a copy-and-paste operation has been performed from a <code>dtterm</code> window, subsequent copy-and-paste operations from the same window to a Windows application always show the data from the first such operation. • Cut-and-paste operations do not work from <code>dtpad</code> to Windows applications. • Cut-and-paste menu options do not work correctly in transfers from StarOffice applications.
Section 17.11, “Compression”	The Windows connector uses RDP bulk compression to compress data between the Sun Ray server, which runs the Windows connector, and the Windows system.
Section 17.9, “Network Security”	The Windows connector uses RSA Security's RC4 cipher, which encrypts data of varying size with a 56-bit or a 128-bit key, to secure all data being transferred to and from the Windows system. Alternatively, using TLS/SSL or CredSSP through the enhanced network security option, all traffic is encrypted as per protocol specifications and system configuration.
Local Drive Mapping	File systems from removable media devices, such as flash drives, can be connected to the Sun Ray Client's USB ports and mapped to the Windows environment, where they are displayed as locally mounted drives. Any file can be mounted and mapped from the Sun Ray environment to the Windows environment. In most cases, USB redirection should be used instead.
Section 17.16, “Printing”	From a Windows session, a user can print from a Windows application using any of the following configurations: a network printer or a locally attached printer on the Windows system, a network printer or a locally-attached printer on the Sun Ray server, or a local printer attached to the Sun Ray Client.
Serial Port Mapping	From a Windows session, users can access external serial devices connected to a Sun Ray Client or an Oracle Virtual Desktop Client running on a Windows client computer.
Section 17.8, “Session Directory”	The Windows connector supports server session reconnection based on load balancing information and the Session Directory, a database that keeps track of which users are

Feature	Description
	running which sessions on which Windows systems. Session Directory functionality enables users to reconnect automatically to the right Windows session. Terminal services session load balancing is handled transparently by the Windows Terminal Server.
Section 17.13, “Smart Cards”	The Windows connector uses the PC/SC framework to enable applications on the Windows system to access smart cards inserted in the desktop client. Typically, this feature is used to provide two-factor authentication with digital certificates or to permit the use of electronic signatures or other information stored on a smart card.

17.2. Requirements

For the list of supported Windows remote desktops, see [Section 3.1.3, “Windows Remote Desktop Support”](#).

17.3. Using the Windows Connector

This section describes how to start a Windows session on a Sun Ray Client or an Oracle Virtual Desktop Client. The Windows connector and the remote Windows system must be configured beforehand. You can configure the Windows connector as part of the Sun Ray Software installation or through the `utconfig -c` command. By default, remote desktop services is not enabled on a Windows system, so you must specifically enable it. See the Windows documentation for details.

The `uttsc` command enables you to establish a remote connection with a Windows system through the Windows connector. An alternative `uttscwrap` command is also provided for users that use JDS on Oracle Solaris.

Many of the functions provided by a local Windows desktop are provided with a Windows session on a desktop client, including the ability to access USB devices connected directly to a Sun Ray Client using USB redirection.

17.3.1. How to Start a Windows Session

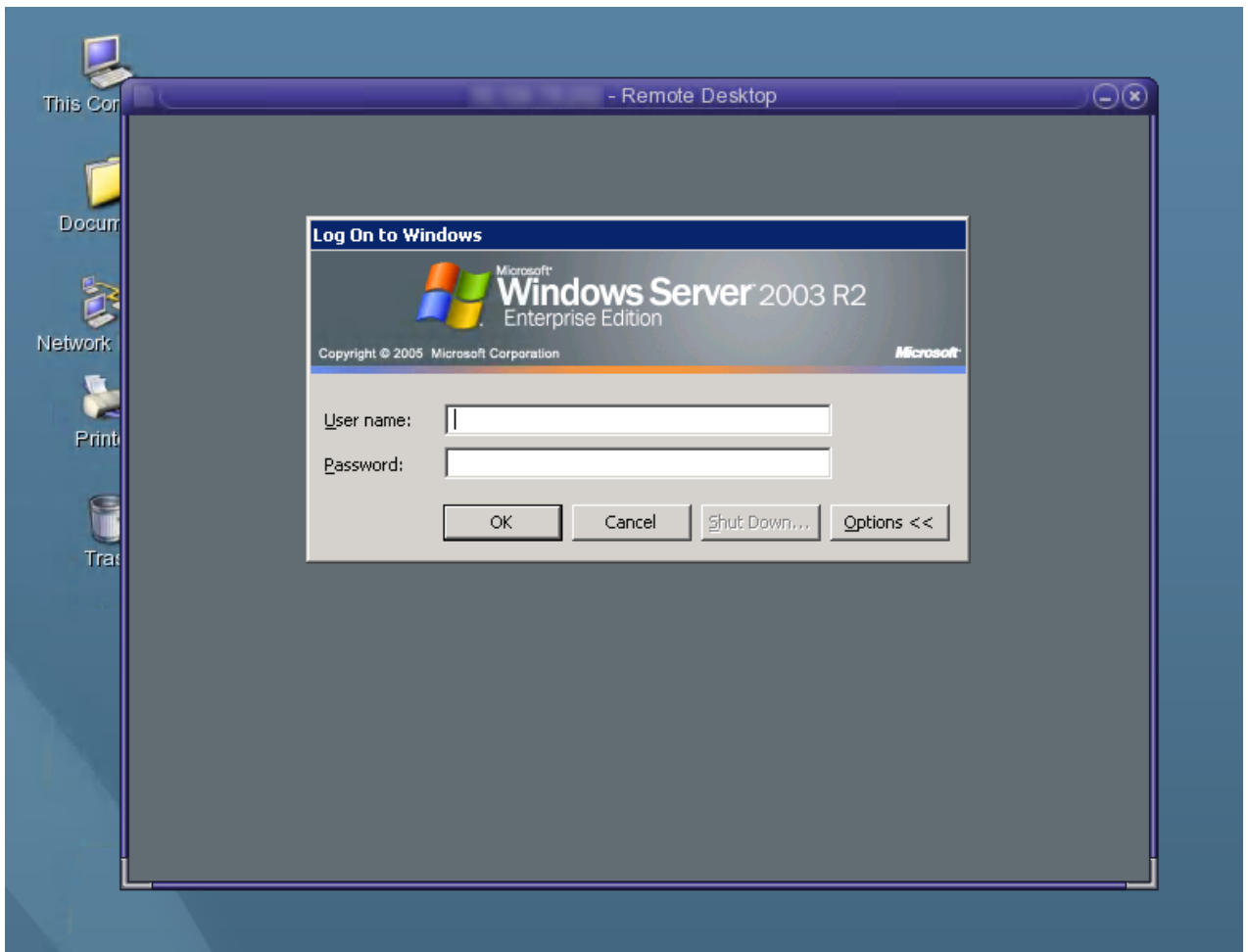
Once the Windows connector has been configured, you can start a Windows session on a desktop client from a Windows system.

1. Log in to a desktop client.
2. Start a Windows session on a Windows system.

```
% /opt/SUNWuttsc/bin/uttsc options hostname.domain
```

If the Windows system is in the same domain as the Sun Ray Client, you do not have to specify the domain name. However, if you prefer, you may specify the full IP address instead of `hostname.domain`.

Issuing the `uttsc` command with no options except the name or address of a Windows system displays a Windows session on the desktop client, as shown in [Figure 17.2, “Windows Connector \(uttsc\) Example”](#).

Figure 17.2. Windows Connector (uttsc) Example

The default screen size is 640 x 480 pixels.

To display a session in full-screen mode or to modify it in other ways, see the `uttsc(1)` man page.

17.3.1.1. Example of `uttsc` Commands

Log in as `user`, enable 24-bit color, set resolution to 1024x768, turn sound quality on high, and connect to the Windows system 192.168.1.20:

```
uttsc -u user -A 24 -g 1024x768 -r sound:high 192.168.1.20
```

Log in as `user`, enable full screen, enable 24-bit color, disables access to the RDP pull-down menu, and connect to the Windows system at 192.168.1.20:

```
uttsc -u user -A 24 -m -b 192.168.1.20
```

Log in as `user`, enable 24-bit color, set resolution to 1024x768, enable sound quality to high, enable 2 factor authentication, and connect to the Windows system at 192.168.1.20:

```
uttsc -u user -A 24 -g 1024x768 -r sound:high -r scard:on 192.168.1.20
```

Log in as `user`, enable 24-bit color, set resolution to 1024x768, enable sound quality to high, map the home directory to Windows H: drive, and connect to the Windows system at 192.168.1.20:

```
uttsc -u user -A 24 -g 1024x768 -r sound:high -r disk:H=path 192.168.1.20
```

Enable full screen session with smart card authentication enabled and connect to the Windows system *windows_system*:

```
uttsc -r scard:on -m windows_system
```

17.3.2. How to Start a Windows Session Within Java Desktop System (Oracle Solaris 10)



Note

The Java Desktop System (JDS) integration package is a deprecated feature and will be removed in a later release.

The Java Desktop System (JDS) integration package for Oracle Solaris 10 delivers the command `uttscwrap` command, which improves integration of the Windows connector with the JDS desktop on Oracle Solaris 10. The JDS integration package is included in the [Supplemental](#) folder of the Sun Ray Software Media Pack. See [Section 3.2.6.1, “How to Install the JDS Integration Package \(Oracle Solaris 10\)”](#) for detailed installation instructions.

To use the `uttscwrap` command, specify the same parameters as the `uttsc` command line. The `uttscwrap` command provides a login dialog that enables you to input credentials for password-based authentication (`username/domain/password`). The credentials can be saved through the dialog for subsequent invocations. At the next launch, the dialog displays the credentials.

Credentials are saved separately for each Windows system and application combination. This convention enables you to save different credentials in the following ways:

- For different applications on the same server
- For different applications on different servers
- For different server sessions with no applications launched

Any new credentials saved for a server or application replace previously saved credentials.



Note

`uttscwrap` is designed for credential caching for password-based authentication only. It cannot be used with smart card authentication. For smart card authentication, use the `uttsc` command.

Steps

1. Log in to a Sun Ray Client.
2. Start a Windows session on a Windows system.

```
% /opt/SUNWuttscwrap/bin/uttscwrap options hostname.domain
```

If the Windows system is in the same domain as the Sun Ray desktop, you do not have to specify the domain name. However, if you prefer, you may specify the full IP address instead of *hostname.domain*.

17.3.3. How to Lock a Windows Session

This procedure describes how to lock a Windows session when a session moves away from a given Sun Ray Client.

**Note**

Implementation of this feature relies on technology not available by default, and it uses non-public Sun Ray interfaces as well as the use of certain public Sun Ray interfaces for purposes other than their intended use. For these reasons, this feature is not provided as a supported feature.

A commonly used approach to implement session locking is to send the lockscreen keystrokes to the Windows Session using `xvkbd`, which is invoked by `utaction`.

You can invoke the `utaction` command from an `Xsession.d` or `xinitrc.d` script as follows:

```
#!/bin/sh
XVKBD=/usr/openwin/bin/xvkdb
/opt/SUNWut/bin/utaction -d "$XVKBD -text 'Ml'" &
```

Because `xvkbd` is not available by default, you should modify the `XVKBD` setting in the example so that it correctly identifies the installation location of `xvkbd`.

**Note**

The keystroke sequence `Ml` activates the Windows lock for Windows XP and Windows Server 2003 R2 sessions. You might need to substitute a different keystroke sequence for other Windows versions.

17.3.4. How to Set Up Access to the `uttsc` Command

To access the `uttsc` command directly, add the following entry to your PATH variable:

```
/opt/SUNWuttsc/bin, /opt/SUNWuttsc/sbin, /opt/SUNWuttscwrap/bin
```

The `/opt/SUNWuttscwrap/bin` path is required only if you are using the JDS integration package.

17.3.5. How to Set Up a Desktop Shortcut to Start a Windows Session

A graphical user interface is not available for the Windows connector at this time. However, you can set up a launcher to create a desktop icon or menu item to connect to the Windows session.

For details about how to set up launchers, consult the desktop documentation for your operating system.

17.3.6. How to Separate Settings for Session Locale and Keyboard Layout

The Windows connector provides the ability to separate settings for the session local and keyboard layout. The `-G` option specifies the language/locale for the session and the `-Y` option specifies the keyboard layout used to process the keyboard input. For example, you can specify the nl-NL Dutch locale with a US international keyboard layout as follows: `uttsc -G nl-NL -Y en-US:INT`.

See the `uttsc` man page for more details. The `-l` option is still available and sets both the language/locale and keyboard layout.

17.4. Audio Input

Audio input redirection connects the audio streams captured on a Sun Ray Client to a Windows session. When a user connects an analog audio input device (such as a microphone or headset) to a Sun Ray

Client, the Windows connector automatically detects and redirects the audio stream to the user's Windows desktop. See [Section 15.6, “USB Headsets”](#) for details on using USB headsets.

Audio input through RDP is not supported for Windows XP or Windows Server 2003 R2, so you must install the audio input Windows connector component. Audio input is automatically provided for Windows 7, Windows Server 2008 R2, Windows 8, and Windows Server 2012.

Audio input redirection for `uttsc` is disabled by default. You can enable audio input by using the following `uttsc` option:

```
-r soundin:[low|medium|high|off]
```



Note

Audio input is not available through VPN connections.

17.4.1. Enabling Audio Input on Windows 7 and Windows Server 2008 R2

Windows 7 and Windows Server 2008 R2 have audio recording redirection disabled by default. Audio recording redirection must be enabled for the audio input feature to work properly.

- To enable audio recording redirection on Windows 7, see [Microsoft knowledge article 2020918](#).
- To enable audio recording redirection on Windows Server 2008 R2, see <http://technet.microsoft.com/en-us/library/dd759231.aspx>.

17.5. Video Acceleration

This section describes the video acceleration features provided with Sun Ray Software. The available features depend on the Windows desktop and the desktop client being used.

Video acceleration improves the user experience for video playback on the client devices. And, by using compressed video streams and client-side decoding and rendering, video acceleration also reduces the use of server CPU cycles and network bandwidth when providing multimedia content.

17.5.1. Video Acceleration Requirements

Video acceleration is available on Windows desktop sessions when the following requirements are met.

Table 17.2. Video Acceleration Requirements

Windows Desktop	Video Type	Required Applications	Required Windows Connector Component	Additional Requirements
Windows 7, 2008, XP, and 2003	Adobe Flash content	32-bit Internet Explorer 8 and 9 Adobe Flash Player	Adobe Flash acceleration component	None
Windows 7 and 2008	MPEG-2, H.264, and VC-1 video	Windows Media Player 12	None (Uses RDP 7 video redirection virtual channel extension)	Audio and video playback in the Remote Desktop Server role must be enabled on Windows Server 2008

Windows Desktop	Video Type	Required Applications	Required Windows Connector Component	Additional Requirements
				R2. See Section 17.5.5, "How to Enable Video Redirection on Windows Server 2008 R2" . When using Oracle Virtual Desktop Client, release 3.2 or later is required.
Windows XP and 2003	MPEG-2, H.264, and VC-1 video	Windows Media Player 10, 11, or 12	Multimedia redirection component	Audio/Video demuxer for MPEG-2 and H.264 videos (for example, MatroskaSplitter). This is required because Windows XP and Windows Server 2003 R2 do not decode MPEG-2 and H.264 natively.

**Note**

For Windows 7 and Windows Server 2008 R2, Sun Ray Software provides an additional heuristic video detection and acceleration algorithm that may benefit other video players. You can enable this additional feature through the `-B on` option of the `uttsc` command. Enabling this feature will not affect the video acceleration feature.

17.5.2. Videos Accelerated

Video acceleration is provided up to a maximum resolution size based on what Sun Ray Client is being used, as specified in the following tables.

**Note**

The maximum resolution sizes listed in the following tables indicate the largest video size that can be used for a reliable user experience on local area networks. The final frame rate provided on the Sun Ray Client is a combination of video content, encoded bitrate, network conditions, upstream software performance, and pipeline efficiency.

Table 17.3. Video Acceleration on Sun Ray 2 Clients

Windows Desktop	Video Type	Profiles	Maximum Resolution Size
Windows 7, 2008, XP, and 2003	Adobe Flash content	n/a	1024x768
Windows 7, 2008, XP, and 2003	MPEG-2	Main Profile Main Level	720x480
	H.264	<ul style="list-style-type: none"> Baseline Profile Level 2.0 Extended Profile minus Data Partitioning 	352x288

Windows Desktop	Video Type	Profiles	Maximum Resolution Size
	VC-1	• Main Profile minus CABAC Entropy Coding	
		Simple Profile Low and Medium Level	352x288
		Main Profile Low, Medium, and High Level	320x240
		Advanced Profile Level 0, 1, 2, and 3	352x288

Table 17.4. Video Acceleration on Sun Ray 3 and 3i Clients

Windows Desktop	Video Type	Profiles	Maximum Resolution Size
Windows 7, 2008, XP, and 2003	Adobe Flash content	n/a	1024x768
Windows 7, 2008, XP, and 2003	MPEG-2	Main Profile Main Level	720x480
	H.264	<ul style="list-style-type: none"> • Baseline Profile Level 2.0 • Extended Profile minus Data Partitioning • Main Profile minus CABAC Entropy Coding 	720x480
	VC-1	<ul style="list-style-type: none"> • Simple Profile Low and Medium Level • Main Profile Low, Medium, and High Level • Advanced Profile Level 0, 1, and 2 	720x480

Table 17.5. Video Acceleration on Sun Ray 3 Plus Clients

Windows Desktop	Video Codec	Profiles	Maximum Resolution Size
Windows 7, 2008, XP, and 2003	Adobe Flash content	n/a	1280x720
Windows 7, 2008, XP, and 2003	MPEG-2	Main Profile Main Level	1280x720
	H.264	<ul style="list-style-type: none"> • Baseline Profile Level 2.0 • Extended Profile minus Data Partitioning • Main Profile minus CABAC Entropy Coding 	1280x720
	VC-1	<ul style="list-style-type: none"> • Simple Profile Low and Medium Level • Main Profile Low, Medium, and High Level 	1280x720

Windows Desktop	Video Codec	Profiles	Maximum Resolution Size
		<ul style="list-style-type: none">Advanced Profile Level 0, 1, and 2	

17.5.3. Audio Accelerated

The following audio formats are decoded to increase audio performance when using video acceleration:

- AAC
- MPEG
- WMA

17.5.4. Additional Notes

Here are some additional notes and restrictions when using video acceleration.

Videos on Windows XP and Windows Server 2003 R2 Desktops

- Some video playback features in Windows Media Player may not work due to decoding video remotely from the host where the player is running. This includes frame-by-frame playback, variable speed playback, repeat mode, the Playlist option, skin mode, moving the slider backward and forward multiple times, and quitting Windows Media Player while playing a video.
- The Mute button and volume slider in the Windows Media Player do not work while playing videos in a Windows session connected from a Sun Ray server running Oracle Linux.
- Multimedia redirection is not supported in a Windows Session Directory environment.
- Third-party video decoders can be installed on the same Windows system as the multimedia redirection component, but the decoders will not be used by the multimedia redirection component when streaming video.
- Creating video content at 15 fps may provide better overall results.
- Pausing a video in Windows Media Player will mute other applications. When resuming the video, the volume level for Windows Media Player and other applications will be set back to the Window Media Player's current volume setting. Any volume level changes made during video playback with the volume keys on the Sun Ray Client are ignored.
- When hotdesking a Windows XP or Windows Server 2003 R2 session while currently playing a video, the audio and video may become out-of-sync or a run-time error may occur. The workaround is to restart Windows Media Player and replay the video. This is a current issue with the MS-RDP client. See [Section 17.5.6.7, "Windows Media Player Error During Session Reconnection"](#) for an example of the Windows Media Player error message.
- Using the Windows Media Player to change volume levels will also change the volume level on other applications. The Windows connector does not provide a per-application volume setting.

Videos on Windows 7 and Windows Server 2008 R2 Desktops

- The Mute button and volume slider in the Windows Media Player do not work while playing videos.
- If you hotdesk a Windows 7 and Windows Server 2008 R2 session that is currently playing video, an error will occur. This may also happen after resetting the Sun Ray Client. See [Section 17.5.6.7, "Windows Media Player Error During Session Reconnection"](#) for details.

- In some instances, Windows 7 and Windows Server 2008 R2 delivers H.264 video packets in the wrong order, which makes the video flicker. If this happens, click the Windows Media Player slider near the current position to fix the problem.
- With high resolution video streams, video play may take a longer time to start than expected. A delay in resuming a video clip may also occur if you change the position of the Windows Media Player slider. The following workaround may resolve these issues. In Windows Media Player, choose **Options** from the Tools menu, click **Buffer** on the Performance tab, and enter **1** in the **seconds of content** field.

Videos on All Windows Desktops

- Video acceleration is not supported in Xinerama sessions (using a single screen across several monitors). For more information about Xinerama, see [Section 17.14, “Multi-Monitor Support”](#).
- Video acceleration is supported only on the primary head of a multihead configuration.
- The volume controller in the Windows task bar cannot be used when playing videos with Windows Media Player.
- Playing video while a copy operation is in progress is not supported.
- Playing video from a USB flash disk attached to a Sun Ray Client is not supported.
- Playing multiple videos simultaneously with Windows Media Player is not supported.
- In some instances, video play may stop. If this happens, click the Windows Media Player slider near the current position to fix the problem.

Adobe Flash Content on All Windows Desktops

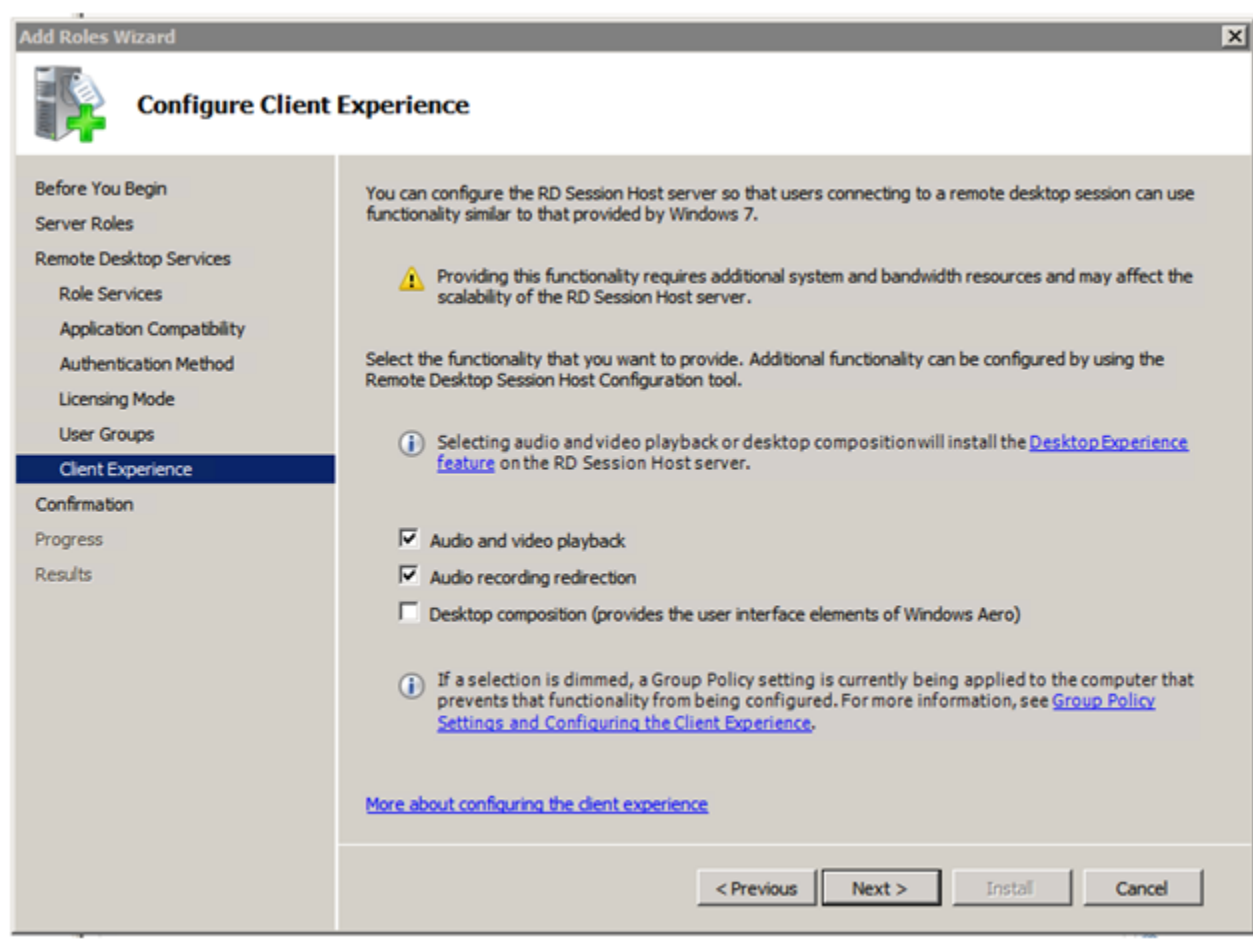
- Playing windowless Adobe Flash content with Adobe Flash Player 11 in Internet Explorer 9 is not supported.
- Adobe Flash content playback performance may not be comparable to a stand-alone desktop environment because of various network latency and lossy video compression issues.
- The audio and video may become out-of-sync during Adobe Flash playback on Windows 7 and Windows Server 2008 R2.
- Adobe Flash acceleration is not supported in Xinerama sessions (using a single screen across several monitors). For more information about using Xinerama, see [Section 17.14, “Multi-Monitor Support”](#).
- Adobe Flash acceleration is supported only on the primary head of a multihead configuration.
- When using 64-bit versions of Windows 7 and Windows Server 2008 R2, a task bar pop-up menu will display behind any Adobe Flash playback area that covers the menu area. If this happens, you need to move the playback area to access the pop-up menu.
- Adobe Flash playback problems may occur when using Adobe Flash Player 11 on Windows Server 2003 R2. Try using Adobe Flash Player 10.x instead.
- Third-party software providing similar media acceleration may conflict with the Flash Acceleration component and make it unusable. To make the Flash Acceleration work properly in this situation, you need to uninstall the third-party software and remove/reinstall the Windows connector components on the Windows system.
- Playing multiple videos simultaneously with Internet Explorer is not supported.

- Some or all control elements of the Adobe Flash settings dialog might not react to mouse or keyboard events, including the button to close the dialog. To workaround this issue, reload the entire web page in the browser.

17.5.5. How to Enable Video Redirection on Windows Server 2008 R2

For Sun Ray Software video acceleration to work from a Windows Server 2008 R2 system, the Windows Server 2008 R2 system must have the Remote Desktop Server role installed and the audio and video playback enabled. For detailed instructions on how to install the Remote Desktop Server role, refer to [Installing and Configuring Remote Desktop Session Host](#). During the installation, select the **Audio and video playback** option on the Configure Client Experience page, as shown in [Figure 17.3, “Enabling Video Redirection on Windows Server 2008 R2”](#).

Figure 17.3. Enabling Video Redirection on Windows Server 2008 R2



17.5.6. Video Acceleration Troubleshooting

This section provides troubleshooting information for the video acceleration feature.

17.5.6.1. How to Enable Video Acceleration Logging

When video acceleration is in use and if logging is enabled, video acceleration status messages and performance statistics are logged in the following files:

- `/var/dt/Xerrors` (Oracle Solaris 10)
- `/var/log/gdm/$DISPLAY.log` (Oracle Linux and Oracle Solaris 11)

Video acceleration logging is disabled by default.

Steps

1. Become superuser on the Sun Ray server.
2. Enable video acceleration logging:

```
# kill -USR2 Xnewt-pid
```



Note

To disable messages, use the same command.

The *Xnewt-pid* value is the *Xnewt* process ID for an individual Sun Ray session.

You can find the *Xnewt* process ID by using the following commands:

For Oracle Solaris:

```
# ps -aef | grep Xnewt | grep userid
```

For Oracle Linux:

```
# /opt/SUNWut/sbin/utsession -p | grep userid
# ps -aef | grep Xnewt | grep "::display"
```

Where *display* is the session display number listed from the *utsession* command.

17.5.6.2. Video Acceleration Status Messages

The video acceleration status messages identify the rendering mechanism used to display the video content on the client. The video acceleration status messages have the following format:

```
Display display-id Video port Id video-port-id stream Id stream-id status-message
```

Here is an example of a status message:

```
Display :3.0 Video port Id 91 stream Id 3 Compressed: H.264
```

You can use the *Video port Id* and *stream Id* values to find the corresponding performance statistics for the video stream, as described in [Section 17.5.6.3, “Video Acceleration Performance Statistics”](#).

In a multihead configuration, *Display* indicates the head on which the video is being played. For example, *Display :3.1* and *Display :3.2*. And, each head's Video port Id is in a different range.

[Table 17.6, “Video Acceleration Status Messages”](#) provides the list of the video acceleration status messages.

Table 17.6. Video Acceleration Status Messages

Message	Comments
Compressed: MPEG-1 Audio	Start of a compressed video stream (multimedia redirection).

Message	Comments
Compressed: MPEG-1/2 Video	
Compressed: VC-1	
Compressed: WMA	
Compressed: H.264	
Compressed: AAC	
Compressed: JPEG-D	Start of a compressed video stream (Adobe Flash acceleration). This message is also provided when improved rendering occurs on Windows 7 and 2008.
Compressed: <i>codec</i> hotdesked firmware does not support compressed video	A compressed video stream tried to connect to a client that does not support decoding, either because of the hardware or outdated firmware.
Compressed: <i>codec</i> error. Replaying headers.	The Sun Ray Client signalled an error and the acceleration feature is resending the video header buffers.
YUV: YV12	Start of an XVideo stream. Note that the XVideo protocol does not require start/stop, so an application may send multiple streams without a new debug message.
YUV: I420	
YUV: YV12 low bandwidth on	An XVideo stream is using the low bandwidth logic or the bandwidth has increased so it is resuming the normal logic.
YUV: YV12 low bandwidth ended	
YUV: YV12 hotdesked or swapped <i>codec</i> hotdesked or swapped	The session running a video stream has been hotdesked.

If Adobe Flash acceleration has occurred (indicated by the video acceleration icon), but there are no status messages in the log file, then the Adobe Flash content was decompressed on the Sun Ray server and displayed through the X11 API.

17.5.6.3. Video Acceleration Performance Statistics

In addition to the general status messages logged for video acceleration, performance statistics for each video stream are logged at approximately one second intervals. The performance statistics can quickly increase the size of the log file by approximately 100 bytes for each second a video runs (or more for larger videos).

Here is an example output of the performance statistics:

```
XvEnc +delta scrn prt strm codec WxH @ X:Y avg fps frames dropped lost overflow ms tpf dtu idle
XvEnc +1.054 :3.0 91 3 JPEG-D(h) 640x390 @ 24:302 26 fps 0 drop 0 lost 0 oflow 18
tpf 145 idle
XvEnc +1.099 :3.0 91 3 JPEG-D(h) 640x390 @ 24:302 28 fps 0 drop 0 lost 0 oflow 3
tpf 238 idle
XvEnc +1.007 :3.0 91 3 JPEG-D(h) 640x390 @ 24:302 31 fps 0 drop 0 lost 0 oflow 3
tpf 202 idle
XvEnc +1.005 :3.0 91 3 JPEG-D(h) 640x390 @ 24:302 28 fps 2 (d) drop 0 lost 0 oflow 21
tpf 209 idle
XvEnc +1.038 :3.0 91 3 JPEG-D(h) 640x390 @ 24:302 27 fps 5 (d) drop 0 lost 0 oflow 38
tpf 151 idle
XvEnc +1.068 :3.0 91 3 MPEG-1/2 Video(se) 352x240 @ 46:96 25 fps 0 drop 0 lost 0 oflow -1
tpf 174 idle
XvEnc +1.009 :3.0 91 3 MPEG-1/2 Video(se) 352x240 @ 54:232 30 fps 0 drop 0 lost 0 oflow -1
tpf 156 idle
```

```

XvEnc +1.001 :3.0 91 3 MPEG-1/2 Video(se) 352x240 @ 54:232 30 fps 0 drop 0 lost 0 oflow -1
tpf 110 idle
XvEnc +1.117 :3.0 91 3 H.264(se) 320x240 @ 70:232 14 (22 d) fps 8 (r) drop 0 lost 0 oflow -1
tpf 10 idle
XvEnc +1.003 :3.0 91 3 H.264(se) 320x240 @ 70:232 15 (31 d) fps 16 (r) drop 0 lost 0 oflow -1
tpf 0 idle
XvEnc +1.103 :3.0 91 3 H.264(se) 320x240 @ 70:232 16 (30 d) fps 14 (r) drop 0 lost 0 oflow -1
tpf 0 idle
XvEnc +1.102 :3.0 91 3 H.264(se) 320x240 @ 70:232 17 (36 d) fps 19 (r) drop 0 lost 0 oflow -1
tpf 0 idle
XvEnc +1.010 :3.0 91 3 VC-1(se) 321x240 @ 70:232 15 fps 0 drop 0 lost 0 oflow -1 tpf 115 idle
XvEnc +1.000 :3.0 91 3 VC-1(se) 321x240 @ 70:232 15 fps 0 drop 0 lost 0 oflow -1 tpf 143 idle
XvEnc +1.000 :3.0 91 3 VC-1(se) 321x240 @ 70:232 15 fps 0 drop 0 lost 0 oflow -1 tpf 125 idle

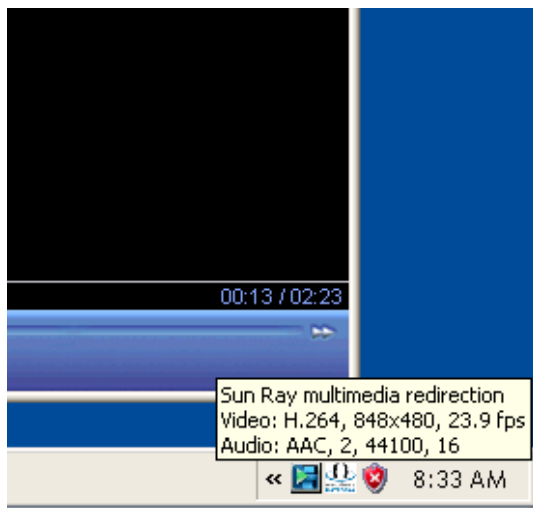
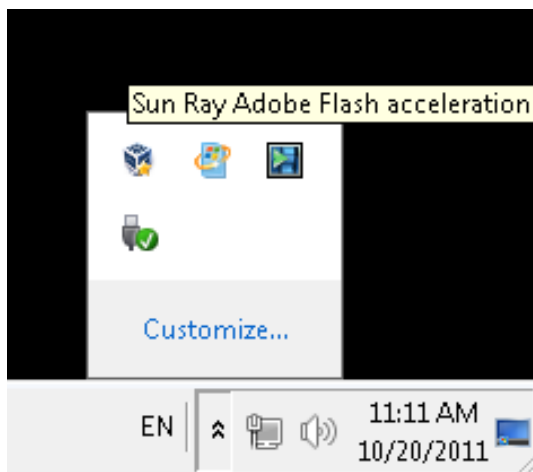
```

The column titles in the output are as follows:

- **XvEnc** - The name of the accelerated video path.
- **+delta** - The delta time (in seconds) from the previous performance entry for the video. This value may be higher (+3) in the first entry when the video is buffering.
- **scrn** - The screen on which the video is displaying.
- **prt** - The video port ID the video is using. Use this value to find the corresponding status message.
- **strm** - The stream ID of the video. Use this value to find the corresponding status message.
- **codec** - The video's codec and whether it is using software or hardware decoding. Values include (**s**) for software decoding without extended firmware, (**h**) for hardware decoding with extended firmware, and (**se**) software decoding with extended firmware.
- **WxH** - The width by height of the video's output size.
- **X:Y** - The X and Y coordinates of the video window's upper-left corner on the screen.
- **avg fps** - The average frames per second for the last measured second of the video stream. Examples include:
 - **21 fps** = 21 fps rendered
 - **17 (26 d) fps** = 26 fps were decoded but only 17 were rendered to the screen
- **frames dropped** - Used by Oracle support.
- **frames lost** - Used by Oracle support.
- **overflow** - Used by Oracle support.
- **ms tpf** - Used by Oracle support.
- **dtu idle** - The average amount of idle time on the Sun Ray Client's CPU. This value ranges from 0-254.

17.5.6.4. How to Verify that Video Acceleration is Active

When video acceleration is active, a video acceleration taskbar icon is displayed in the Windows desktop taskbar. It is displayed on Windows XP and Windows Server 2003 R2 with multimedia redirection enabled and on all Windows platforms with Adobe Flash acceleration enabled. Hovering the mouse over the icon displays the type of acceleration, the media type, and size, as shown in the following screenshots.

Figure 17.4. Verifying Video Acceleration is Active (Multimedia Redirection)**Figure 17.5. Verifying Video Acceleration is Active (Adobe Flash Acceleration)**

17.5.6.5. Adobe Flash Acceleration Not Working (Icon Not Displaying)

Check the following:

- Make sure the Adobe Flash acceleration Windows component is installed on the Windows system. See [Section 3.2.7, "How to Install the Windows Connector Components on a Windows System"](#) for the details.
- Make sure that Internet Explorer has the "Third party browser extensions" option enabled, which is located in the Advanced tab of **Tools > Internet Options**.

17.5.6.6. Audio/Video is Out of Sync When Playing VC-1 Movies

If the video acceleration play icon tool tip shows Audio:NONE instead of Audio:PCM, the audio device for the Windows Media Player is not configured properly.

Workaround

Delete the following Windows registry key to revert back to the default audio device for the Windows Media Player:

```
HKEY_CURRENT_USER/Software/Microsoft/MediaPlayer/Preferences/DefaultAudioDevice
```



Caution

Always back up the registry on the Windows system before modifying registry keys.

17.5.6.7. Windows Media Player Error During Session Reconnection

If a Windows connector session is relaunched or hotdesked while a supported media format clip is playing, a Windows Media Player error alert box might be displayed, as shown in [Figure 17.6, “Windows Media Player Error During Session Reconnection”](#).

Figure 17.6. Windows Media Player Error During Session Reconnection



Workaround

Restart Windows Media Player and replay the video.

17.5.6.8. How to Disable Multimedia Redirection

By default, multimedia redirection for Windows XP and Windows Server 2003 R2 is enabled when using the `uttsc` command. The standard RDP protocol is used for video and audio when the multimedia redirection feature is disabled.

- To disable multimedia redirection, use the `-M off` option when issuing the `uttsc` command.

```
% uttsc -M off other_uttsc_options
```

17.5.6.9. How to Disable Adobe Flash Acceleration

By default, Adobe Flash acceleration is enabled when using the `uttsc` command. The standard RDP protocol is used for Adobe Flash content when the Adobe Flash acceleration feature is disabled.

- To disable Adobe Flash acceleration, use the `-F off` option when issuing the `uttsc` command.

```
% uttsc -F off other_uttsc_options
```

17.6. USB Device Redirection

The USB redirection feature enables users to access USB devices connected to a Sun Ray Client from their Windows sessions, provided that the appropriate device drivers are installed on the Windows system.

Once you install the USB redirection component and add USB Drivers to the Virtual Machines (when needed), users can simply plug in and access the USB devices from their Sun Ray Client.

**Note**

Human Interface Devices (HID) such as keyboards and mice do not use the USB redirection component.

The USB device service is enabled by default on the Sun Ray server, which enables the USB redirection functionality to work on the clients. You can use the `utdevadm` command or the **Advanced > Security** page on the Admin GUI to check if the USB device service is enabled.

17.6.1. Device Access

The accessibility of USB devices through USB redirection is determined by what Windows operating system you are using for the remote desktop connection.

When using the single-user Windows XP, Windows 7, and Windows 8 platforms, the USB devices connected to a client are accessible only to the user logged into the client's Windows session.

When using the Windows Server 2003 R2, Windows Server 2008 R2, and Windows Server 2012 platforms, the USB devices connected to a client are accessible and visible to all desktops running on the Windows system. Sharing USB devices between multiple clients does not require any additional setup. Users are always prompted to verify if it is acceptable to share their USB device with others.

17.6.2. Supported Configurations

For the list of supported Windows operating systems, see [Section 3.1.3, “Windows Remote Desktop Support”](#).

USB redirection is available through the following configurations:

- Windows connector in kiosk mode
- VMware View connector in kiosk mode

17.6.3. Tested USB Devices

For the list of USB devices tested to work with the USB redirection feature, see the [Sun Ray Client and Oracle Virtual Desktop Client Peripherals](#) document on the Oracle Technology Network.

**Note**

USB headsets do not require or use USB redirection. See [Section 15.6, “USB Headsets”](#) for details.

17.6.4. Additional Notes

- Devices should be connected to a user's session only after a Windows session is established. When users exit their session, the device should be disconnected.
- If a device is connected before a Windows session is established and the device is not available in the Windows session, hotplugging the device will make it available to the Windows session.
- Before disconnecting a USB device being used through USB redirection during a live Windows session, users must follow the same steps to safely remove the USB device as if the device were directly connected to Windows.

- There is no limit to the number of USB devices that USB redirection can support on a client. A USB hub can be used to expand the number of physical USB ports if needed.
- CCID-compliant USB smart card readers do not use USB redirection. Instead, they use the RDP smart card channel if the `-r scard:on` option of the `uttscc` command is specified. USB smart card readers that are not CCID-compliant use USB redirection, but they cannot be used for Windows session authentication.
- The following scenarios might lead to data corruption on the device:
 - Hotplugging a device during data transfer
 - Hotdesking during data transfer
 - If the session is disconnected for any reason
- Some unpowered USB devices may draw more current than what is supported by the Sun Ray Client. If you see the icon shown in [Figure 17.7, “USB Redirection Overcurrent Icon”](#), then the device may not work properly.

Figure 17.7. USB Redirection Overcurrent Icon



- Writing files to USB flash drives with Oracle Linux may take longer than expected. This reduced performance is because of the 1 Kbyte block size and the file synchronization mechanism of Oracle Linux.
- Writing files to USB secure flash drives may require administrator permissions on the Windows system.

17.6.5. How to Add USB Drivers to a Virtual Machine

If your Virtual Machine (VM) does not have the USB driver installed by default, you must install the driver for USB device redirection to work properly. Examples of VMs that require this step include VMWare ESX and Hyper-V Server.

This procedure should be done before the USB redirection feature is installed. For details on installing the USB redirection feature, see [Section 3.2.7, “How to Install the Windows Connector Components on a Windows System”](#).

1. Make sure the Windows system has access to the Windows XP ISO used to create the VM.
2. Copy the `usbd.sy_` file from the Windows XP ISO to the VM.

For 32-bit:

```
cp ISO-image\i386\usbd.sy_ \windows\system32\drivers
```

For 64-bit:

```
cp ISO-image\amd64\usbd.sy_ \windows\system32\drivers
```

3. Change to the `drivers` directory.

```
cd \windows\system32\drivers
```

4. Install the USB drivers.

```
expand usbd.sy_ usbd.sys
```

5. Reboot the VM.

17.6.6. USB Redirection Troubleshooting

Here is a list of questions if USB redirection is not working.

- Is the [USB redirection taskbar icon displayed](#) in the Window's System Tray.
- Are you using a tested USB device? Check the [Sun Ray Client and Oracle Virtual Desktop Client Peripherals](#) document for the list of tested USB devices.
- Are you using USB redirection in a supported configuration. See [Section 17.6.2, "Supported Configurations"](#) for details.
- Is the USB device service enabled on the Sun Ray server? Use the `utdevadm` command or the **Advanced > Security** page on the Admin GUI to check if the USB device service is enabled.
- Are you using a USB device that draws more current than what is supported by the Sun Ray Client. If there is a power problem with the device, the overcurrent icon is displayed, as shown in [Figure 17.7, "USB Redirection Overcurrent Icon"](#).
- Was the USB device was plugged in *after* the Windows session was established? If not, disconnect the USB device and plug it in again.
- Are you using a USB external smart card reader for session authentication. If so, this will produce unpredictable behaviors.
- Is the device driver for the USB device configured properly in the Windows Device Manager?

If the device driver is not configured properly, a yellow question mark is displayed next to the USB device's entry in the Device Manager. You must install the device driver for the USB device on the Windows system.

- Are the USB redirection device drivers (`utSrServerBus` and `utSrDtuBus`) configured properly in the Windows Device Manager?

[Figure 17.8, "Verifying USB Redirection in Windows Device Manager"](#) shows that the USB redirection component is configured properly (`utSrServerBus` under Device Manager/System devices and `utSrDtuBus` under Device Manager/Universal Serial Bus controllers). If they are missing, you must reinstall the USB redirection component. See [Section 3.2.7, "How to Install the Windows Connector Components on a Windows System"](#) for details.

Figure 17.8. Verifying USB Redirection in Windows Device Manager

17.6.6.1. How to Verify that USB Redirection is Active

When USB redirection is active and running in a session, a USB redirection taskbar icon is displayed in the Windows desktop taskbar, as shown in [Figure 17.9, "Verifying USB Redirection is Active"](#).

Figure 17.9. Verifying USB Redirection is Active

When you see this icon, you can connect USB devices to the Sun Ray Client.

If you don't see this icon but you know the component is installed, run the following command on the Windows system to restart the USB redirection component:

```
C:\Program Files\Oracle\Sun Ray\utUsbRedirector\utUsbRedirector.exe
```

17.6.6.2. USB Redirection Log File

By default, the `/var/opt/SUNWut/log/uttscpd.log` log file provides some USB redirection error messages. To enable full debug information for USB redirection, perform the following steps:

1. Become superuser on the Sun Ray server.
2. Uncomment the `USB_DEBUG_ON` variable in the `/etc/init.d/uttscp` file and make sure it is set to something like `"-D 20"`

```
USB_DEBUG_ON="-D 20"
```

3. Restart the Windows connector proxy daemon.

```
# /opt/SUNWuttsc/sbin/uttscrestart
```

17.7. Hotdesking

This section describes the various ways you can manage a Windows session when hotdesking occurs.

17.7.1. Hotdesking Behavior

The `uttsc` command enables you to specify what happens to the Windows session when a user hotdesks to another client. You can set this behavior by using the `-H` option.

The modes include:

- `-H reconnect` - If the remote desktop server is configured in Device Client Access License Mode, hotdesking a Sun Ray session disconnects and reconnects the existing remote desktop session. The user must re-enter credentials. This is the default mode.
- `-H nodisconnect` - Hotdesking the Sun Ray session does not disconnect and restart the remote desktop session. The remote desktop session remains connected. Previously the `-O` option.
- `-H autoreconnect` - If the Automatic Reconnection feature is enabled on the Windows remote desktop server, hotdesking a Sun Ray session always reconnects to the remote desktop session. This action updates the client information on the remote desktop server. The user does not have to re-enter credentials. See [Section 17.10, "Automatic Reconnection"](#) for more information.

17.7.2. Location Awareness

Location awareness is a feature that provides additional hotdesking capabilities for a Windows session, which enables you to:

- Obtain the unique client's name in a Windows session after session startup or even after hotdesking. The client's name is forwarded during hotdesking.
- Set up actions through commands or scripts to execute in a Windows session when the associated client session disconnects and reconnects during hotdesking. Actions set up for reconnection also occur at session startup.



Note

In the context of the Windows connector, the client's name is the client ID of the Sun Ray Client or Oracle Virtual Desktop Client, also known as the DTU ID. See [Section 13.1.1, "Client ID Differences Between Oracle Virtual Desktop Clients and Sun Ray Clients"](#) for more details.

In some situations, this feature replaces the need to use the `utaction` at the Sun Ray server operating system level, such as setting up follow-me printing.

Location awareness sets several environment variables, which can be used when actions are executed in a Windows session. [Table 17.7](#) lists the environment variables and the information they contain.

Table 17.7. Location Awareness Environment Variables

Environment Variable	Description
<code>UTCINFO_CLIENTIPA</code>	IP address of the Sun Ray Client or Oracle Virtual Desktop Client.
<code>UTCINFO_CLIENTNAME</code>	The client ID of the Sun Ray Client or Oracle Virtual Desktop Client.
<code>UTCINFO_CLIENTLOCATION</code>	The location of the client, as defined by a Sun Ray server administrator. See Section 13.1.6, "How to Configure a Client's Location and Information" for details.
<code>UTCINFO_CLIENTOTHERINFO</code>	Other information about the client, as defined by a Sun Ray server administrator. See Section 13.1.6, "How to Configure a Client's Location and Information" for details.

When a session starts or reconnects, the value of the `UTCINFO_CLIENTNAME` variable is copied to the Windows `CLIENTNAME` environment variable and the `HKey_Current_User\Volatile Environment\CLIENTNAME` registry key.

To enable location awareness, you must install the Client Information Agent on the Windows system using the Sun Ray Windows Components installer, which is described in [Section 3.2.7, “How to Install the Windows Connector Components on a Windows System”](#). Once installed, location awareness is enabled by default and is automatically used when a Windows session starts.

17.7.2.1. Obtaining a Client's Name in a Windows Session

The location awareness feature enables you to obtain a client's name after session startup or even after hotdesking. The client name can be used for various configuration scenarios. You can obtain a client's name by using one of the standard Windows interfaces:

- The `CLIENTNAME` environment variable.
- The `HKCU\Volatile Environment\CLIENTNAME` registry key.
- Using the `GetComputerName()` function in a Windows desktop session.
- Using the `WTSSessionQueryInformation()` function in a Terminal Services session.

17.7.2.2. Setting Up Actions for a Windows Session

The location awareness feature enables you to set up commands or scripts to execute in a Windows session when the associated client session disconnects or reconnects during hotdesking, and when a Windows session starts. To do this as administrator, specify one or more registry values, *name=data* pairs, to the following registry keys:

For 32-bit systems

- For session disconnects - `HKLM\Software\Oracle\Sun Ray\ClientInfoAgent\DisconnectActions`
- For session reconnects and session startup - `HKLM\Software\Oracle\Sun Ray\ClientInfoAgent\ReconnectActions`

For 64-bit systems

- For session disconnects - `HKLM\Software\Wow6432Node\Oracle\Sun Ray\ClientInfoAgent\DisconnectActions`
- For session reconnects and session startup - `HKLM\Software\Wow6432Node\Oracle\Sun Ray\ClientInfoAgent\ReconnectActions`



Caution

Always back up the registry on the Window system before modifying registry keys.

Here are some examples of registry values for the registry keys mentioned above, where the `Commandn` name is used to imply order.

```
Command1=notepad.exe
Command2=wscript.exe c:\tmp\myscript.vbs
```

The *data* value specifies the command or script to be executed, and you can specify either a *'String'* or `REG_SZ` value type.

For an executable command, such as a `.exe` file, you can specify an absolute path. If you do not provide a path, the executable is searched for in the following order: the current directory, the Windows system directory, the Windows directory, and the directories in the PATH environment.

For a script, you should specify the script to be run in an interpreter or shell and the script path must be absolute. For example `cmd.exe /c c:\foo\script.bat` or `wscript.exe c:\foo\script2.vbs`

17.7.2.3. Location Awareness Examples

There are a number of ways to use location awareness for real-world situations. Here are just a few examples.

- A health care provider requires access to the local printer in each patient's room. By using the `ReconnectActions` registry key, you can specify a script to run whenever a health care provider logs in to a room's Sun Ray Client. For this situation, you would need to create a script to read the new client's name (Sun Ray Client's unique ID), perform a lookup to determine the printer in the room, and configure the Windows session's default printer to be the room's printer. You could also use the `DisconnectActions` registry key to run another script that removes the currently configured printer when the health care provider disconnects from the Sun Ray Client.
- An instructor wants to automatically display the student's daily syllabus and lab instructions when the students log in. Since you know the client's name of the Sun Ray Clients in the training room, you can set up a script to automatically display the training content when the students log in to any of the Sun Ray Clients in the classroom. Again, this script would be run by the `ReconnectActions` registry key.

17.8. Session Directory

The Windows connector supports server session reconnection based on load-balancing information and the Session Directory, a database that keeps track of which users are running which sessions on which Windows system. Session Directory functionality enables Windows connector users to reconnect automatically to the right Windows session.

Both IP address-based and token-based reconnection are supported. However, token-based redirection requires the use of a hardware-based load balancer for Windows systems configured as a server farm. The capacity to use server farms and load balancing enables Windows systems to accommodate a larger number of Sun Ray users and clients.



Note

To participate in a Session Directory-enabled server farm, the Windows systems must run Windows Server 2003 R2, Windows Server 2008 R2, or Windows Server 2012. Session Directory is an optional component that can be configured to use Microsoft proprietary or third-party load balancing products.

Terminal services session load balancing is handled transparently by the Windows Terminal Server.

For details about setting up and managing a Session Directory and load balancing, refer to the [Microsoft documentation](#).

17.9. Network Security

To secure all data being transferred to and from the Windows server, the Windows connector supports built-in RDP network security and enhanced network security options. The built-in RDP security uses the RC4 cipher, which encrypts data of varying size with a 56-bit or a 128-bit key. The enhanced network security options include TLS/SSL (with optional server verification) and Network Level Authentication (NLA) using CredSSP.

17.9.1. Built-in RDP Network Security

The Windows connector uses RSA Security's RC4 cipher to secure all data being transferred to and from the Windows system. This cipher encrypts data of varying size with a 56-bit or a 128-bit key.

Table 17.8, “Encryption Levels for Network Security” lists the four levels of encryption that can be configured on the Windows system.

Table 17.8. Encryption Levels for Network Security

Level	Description
Low	All data from client to server is encrypted based on maximum key strength supported by the client.
Client-compatible	All data between client and server in both directions is encrypted based on the maximum key strength supported by the client.
High	All data between the client and server in both directions is encrypted based on the server's maximum key strength. Clients that do not support this strength of encryption cannot connect.
FIPS-Compliant	FIPS-compliant encryption is not supported.



Note

Data encryption is bidirectional except at the Low setting, which encrypts data only from the client to the server.

17.9.2. Enhanced Network Security

The enhanced network security options include TLS/SSL (with optional server verification) and Network Level Authentication (NLA) using CredSSP. These options protect the Windows session from malicious users and software before a full session connection is established.

Table 17.9, “Command Line Examples for Enhanced Network Security” provides a list of `uttscc` command line examples that show which security mechanism is used when the Windows remote desktop service is configured to negotiate with the client. A result of "RDP" means that the built-in RDP security is used.

Table 17.9. Command Line Examples for Enhanced Network Security

uttscc Command Line Examples	Windows XP	Windows Server 2003 R2	Windows 7, Windows Server 2008 R2	Windows 8, Windows Server 2012
<code>-u user -p</code>	RDP	SSL/TLS	NLA	NLA
<code>-u user -j VerifyPeer:on</code>	RDP	SSL/TLS	SSL/TLS	SSL/TLS
<code>-u user -j VerifyPeer:on -p</code>	RDP	SSL/TLS	NLA	NLA
<code>-N off</code>	RDP	RDP	RDP	RDP

17.9.2.1. TLS/SSL Security

Use the following notes to configure TLS/SSL security:

- The RDP host must be running Windows Server 2003 R2, Windows 7, Windows Server 2008 R2, Windows 8, or Windows Server 2012.

- The Windows system's security layer must be configured as "SSL (TLS 1.0)" or "Negotiate."
- In order to connect to a Windows host with TLS/SSL peer verification enabled (`-j VerifyPeer:on`), you must add the root certificate to the client's OpenSSL cert store or specify an additional search path/ PEM file by using the `-j CAPath:path` or `-j CAfile:pem-file` options of the `uttsc` command.
- TLS/SSL connections require a certificate to be present on the Windows system. If that is not the case, the connection might fall back to the built-in RDP security (if allowed) or fail.

17.9.2.2. NLA Security

Use the following notes to configure NLA security:

- The RDP host must be running Windows 7, Windows Server 2008 R2, Windows 8, or Windows Server 2012.
- The Windows system's security layer must be configured as "SSL (TLS 1.0)" or "Negotiate."
- You can enforce NLA security on a Windows system. For example, when using Windows Server 2008 R2, select the following option on the Remote tab of the System Properties window: "Allow connections only from computers running Remote Desktop with Network Level Authentication (more secure)". With this option selected, users must use the `-u` and `-p` options with the `uttsc` command to connect to the server.
- By default, Kerberos authentication is used for NLA security. If the Windows connector cannot obtain a Kerberos ticket for the remote desktop service, it will use NT LAN Manager (NTLM) authentication. You can specify the `-N kerberos` option of the `uttsc` command to force it to use only Kerberos authentication or the `-N ntlm` option to force it to use only NTLM authentication.
- When using Kerberos authentication, the user name specified must be a valid Kerberos principal name. Use the `kinit` command to verify if a principal name is valid.
- When using Kerberos authentication, Kerberos must be configured on the Sun Ray server as described in [Section 17.9.2.3, "Setting Up Kerberos Authentication for NLA Security"](#).

17.9.2.3. Setting Up Kerberos Authentication for NLA Security

If you want to use Kerberos authentication for NLA security, Kerberos must be configured properly on the Sun Ray server. Refer to the following information about setting up Kerberos authentication:

- `krb5.conf(4)` man page - <http://docs.oracle.com/docs/cd/E19253-01/816-5174/6mbb98ufn/index.html>
- Kerberos Service on Oracle Solaris - <http://docs.oracle.com/docs/cd/E19253-01/816-4557/seamtm-1/index.html>
- Kerberos on Oracle Linux - http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/5/html/Deployment_Guide/ch-kerberos.html

Once configured, you can check that Kerberos and its name resolution requirements are configured properly by using the `getent`, `nslookup`, and `kinit` commands.

For example:

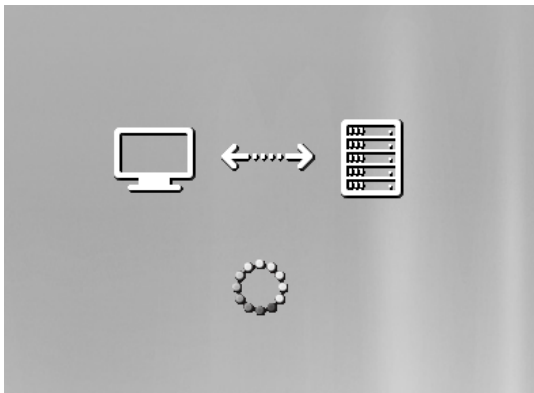
- `# getent hosts my.windows.host` must return the IP address and the hostname
- `# getent hosts IP_of_my.windows.host` must return the IP address and the hostname
- `# nslookup -query=any _gc._tcp.domain_name` must resolve the domain

- `# kinit -V super-user@domain_name` must succeed

17.10. Automatic Reconnection

If the Automatic Reconnection feature is enabled on the Windows remote desktop server, the Windows connector automatically re-establishes a network connection if it is unexpectedly disconnected. When you are disconnected from a Window session, the Windows connector displays an icon on your screen, as shown in [Figure 17.10, “Automatic Reconnection Icon”](#).

Figure 17.10. Automatic Reconnection Icon



By default, the Windows connector attempts to reconnect six times before ending the connection. You can control the number of reconnects through the `-U number` option of the `uttsc` command. Specifying `-U 0` disables the use of the Automatic Reconnection feature.



Note

You can also use the `-S timeout` option, in seconds, to specify the interval to detect network loss for a reconnect to occur.

17.11. Compression

The Windows connector uses RDP bulk compression to compress data between the Sun Ray server, which runs the Windows connector, and the Windows system.

Compression is enabled by default.

17.11.1. How to Disable Compression

You can be disable compression on a per-connection basis.

To disable compression, use the `-z` option of the `uttsc` command.

17.12. Licensing

Microsoft Terminal Services licensing information is stored in the Sun Ray data store automatically upon Windows session startup, using the existing LDAP schema. No administrator setup or intervention is required.

Licenses can be administered, such as listing and deleting licenses, with the `utlicenseadm` command. See the `utlicenseadm` man page for details.

The Windows connector supports both per-user and per-device Terminal Server Client Access Licenses (TS-CAL):

- **Per-user mode** - The user's hotdesking experience is virtually seamless.
- **Per-device mode** - The user must reauthenticate every time they hotdesk to a different client to ensure correct TS-CAL license handling.



Note

If you access terminal server functionality provided by Microsoft operating system products, you need to purchase additional licenses to use such products. Consult the license agreements for the Microsoft operating system products that you are using to determine which licenses you must acquire. Currently, information regarding Terminal Services can be found at: <http://www.microsoft.com/windowsserver2008/en/us/how-to-buy.aspx>

17.12.1. Per-user Mode Versus Per-device Mode

To show the different behavior between the per-user and per-device modes, let's start with the user logging into a Sun Ray session with a smart card and opening a connection to a Windows session. [Table 17.10, "Windows Licensing Modes"](#) shows what happens next when the user removes the smart card and inserts it again.

Table 17.10. Windows Licensing Modes

The User Removes the Smart Card and...	Per-user Mode	Per-device Mode
Reinserts the Smart Card in the same client.	The user is instantly reconnected to the existing Windows session.	The user is instantly reconnected to the existing Windows session.
Inserts the Smart Card in a different client.	The user is instantly reconnected to the existing Windows session.	<p>The Windows login screen prompts the user for username and password, after which the user is reconnected to the existing Windows session. Other features and services are similarly affected. For example:</p> <ul style="list-style-type: none"> • Windows Media Player stops playing audio/video file, although the application is still active on the Windows session. The user needs to replay the audio/video file. • Any serial port transfer is stopped. All the command line options specified remain valid.



Note

You can use the `-H nodisconnect` option of the `uttsc` command to prevent the Windows connector from disconnecting upon detection of hotdesking events.



Note

With the `-H nodisconnect` option, the Windows connector does not disconnect and reconnect when a hotdesk event occurs, nor does it refresh licenses on different clients. Instead, it uses the original license granted upon connection to the first client. This behavior might cause you to inadvertently violate your Microsoft Terminal Server license agreement. Because you have full responsibility for license

compliance, be aware of the danger and use the `-H nodisconnect` option only with caution.

17.13. Smart Cards

In addition to normal Sun Ray smart card functionality, such as hotdesking, the Windows connector enables additional smart card functionality, such as the following:

- Strong, two-factor authentication for access control with digital certificates
- PIN-based logins
- Digital signing, encrypting, and decrypting of email messages from Windows-based email clients

The Windows connector uses the smart card services on the Sun Ray server and smart card middleware on the Windows system. For detailed information about the smart card services and the configuration details, see [Chapter 8, Smart Card Services](#).

If you want to use an external smart card reader for Windows session authentication, a CCID-compliant USB smart card reader must be used. See [Section 17.13.1, “How to Enable Smart Card Readers on a Windows System”](#) for details.

17.13.1. How to Enable Smart Card Readers on a Windows System

This section describes how to redirect a smart card reader connected to a desktop client so it can be used by the Windows system. The smart card services on the Sun Ray server must still be configured, which is described in [Section 8.6, “Configuring Smart Card Services”](#).

To enable internal or external smart card readers, you must use the `-r scard:on` option of the `uttsc` command.

CCID-compliant USB smart card readers are redirected through the Windows RDP smart card channel, which enables the smart card to be used for Windows session authentication. USB smart card readers that are not CCID-compliant use USB redirection, but they cannot be used for Windows session authentication.

17.13.2. How to Set Up Smart Card Login for Windows

This procedure describes how to set up smart card login for Windows.

1. Set up Active Directory and Certification Authority (CA) on the Windows system.
2. Install the smart card middleware product on the Windows system.



Note

If you use ActivClient middleware, set the Disable PIN Obfuscation option to Yes through the ActivClient user console on the Windows system.

3. Enroll the necessary certificates onto the smart card using either a Sun Ray token reader or an external smart card reader connected to the Windows system.

17.14. Multi-Monitor Support

Sun Ray Software supports specific multiple monitor configurations as described in [Chapter 12, Multiple Monitor Configurations](#). When using the Windows connector, the following monitor configurations are

supported by the multi-monitor feature on a Windows 7, Windows Server 2008 R2, Windows 8, and Windows Server 2012 remote desktop session:

- A multi-monitor configuration on a Sun Ray Client with dual video connectors, specifically a Sun Ray 2FS Client or Sun Ray Client 3 Plus Client. You must specify the `-X xrandr` option of the `uttsc` command.
- A multi-monitor configuration with Oracle Virtual Desktop Clients. The default `-X xinerama` option of the `uttsc` command is used.
- A multihead group configuration with Xinerama enabled. Two monitors connected to Sun Ray 2FS Clients or Sun Ray 3 Plus Clients in a multihead group will be enumerated to Windows as a single monitor with the combined dimensions of the two monitors. The default `-X xinerama` option of the `uttsc` command is used.

**Note**

When playing a video in the primary display, the video will stop playing if you drag the video window to the secondary display. This is a limitation in Windows.

The Xinerama enumeration is enabled by default for the Windows connector. To disable this feature, use the `-X off` option of the `uttsc` command. Refer to the `uttsc` man page for more information about the `-X` option.

For detailed information about how to configure the remote desktop session to use multiple monitors, see [Configure Monitor Settings for a Remote Session](#).

The supported configurations for multihead group size and geometry are the same when using a Windows session. If you want to span the remote desktop session across multiple monitors, you can use the full screen mode (`-m` option of the `uttsc` command) together with a multihead group.

17.15. Dynamic Session Resizing

Dynamic session resizing allows the remote desktop to be resized automatically to fit the optimized size of your local desktop client session. See [Section 13.1.2, “Dynamic Session Resizing”](#) for details on enabling this feature.

If dynamic session resizing is enabled, the Windows desktop will automatically resize if you use the `-f all` option (full screen mode) or the `-g` option (window mode) of the `uttsc` command.

17.16. Printing

The Windows connector supports printing for the following printer configurations:

- Network printers visible on the Windows system
- Network printers visible on the Sun Ray server
- Local printers attached to the Windows system
- Local printers attached to the Sun Ray server
- Local printers attached to the client

Here are some important notes about setting up printers for the Windows connector.

- Network printers are not affected by hotdesking. Printers connected to clients are available for printing from any client connected to the same Sun Ray server.
- For printers accessible through the Sun Ray server (network visible or local), you need to perform some initial configuration to make the printers accessible through the Windows connector.

17.16.1. How to Set Up Print Queues (Oracle Solaris 10)

This procedure describes how to set up a raw print queue on a Sun Ray server running Oracle Solaris 10 so that a Windows system can access it. This procedure is typically needed for printers locally attached to the Sun Ray server.

If a network printer is visible on the Sun Ray server, this typically indicates that the queue has been set up already and you should not have to perform this task. These instructions pertain to raw print queues, which are print queues configured without a printer driver. Please consult your operating system documentation for instructions about setting up queues for PostScript drivers. See also the [lp](#) and [lpadmin](#) man pages.

1. Specify the printer and printer device node using the [lpadmin](#) command.

```
# /usr/sbin/lpadmin -p printer-name \  
-v /tmp/SUNWut/units/IEEE802.mac-address/dev/printers/device-node
```

2. Enable the print queue.

```
# /usr/bin/enable printer-name
```

3. Accept the print queue.

```
# /usr/sbin/accept printer-name
```

To update the Windows session with the available print queues on the Sun Ray server, you must restart the Windows connector with the relevant print queues specified on the command line. See [Section 17.16.3](#), “How to Make Sun Ray Printers Available to a Windows Session” for details.

17.16.2. How to Set Up Print Queues (Oracle Linux)

This procedure describes how to set up a raw print queue on a Sun Ray server running Oracle Linux, so that it can be accessed by a Windows system. This procedure is typically needed for printers locally attached to the Sun Ray server.

If a network printer is visible on the Sun Ray server, the queue has been set up already and you should not have to perform this task. These instructions pertain to raw print queues, which is a print queue configured without a printer driver. Please consult your operating system documentation for instructions on setting up queues for PostScript drivers. See also the [lp](#) and [lpadmin](#) man pages.

1. Uncomment the following line from the [/etc/cups/mime.convs](#) file:

```
application/octet-stream application/vnd.cups-raw 0 -
```

2. Uncomment the following line from the [/etc/cups/mime.types](#) file:

```
application/octet-stream
```

3. Restart the [cups](#) daemon.

```
# /etc/init.d/cups restart
```

4. Create a soft link to the Sun Ray printer node in [/dev/usb](#)

For example, if the device node is `/tmp/SUNWut/units/IEEE802.mac-address/dev/printers/device-node`, then use the following command:

```
# ln -s /tmp/SUNWut/units/IEEE802.mac-address/dev/printers/device-node \
/dev/usb/sunray-printer
```

Use this soft link (`/dev/usb/sunray-printer`) as the Device URI while creating the print queue.



Note

After rebooting, you might have to create the `/dev/usb` directory and re-create the soft link.

5. To complete the procedure, set up a raw print queue.

```
# /usr/sbin/lpadmin -p printer-name -E -v usb:/dev/usb/sunray-printer
```

To update the Windows session with the available print queues on the Sun Ray server, you must restart the Windows connector with the relevant print queues specified on the command line. See [Section 17.16.3, “How to Make Sun Ray Printers Available to a Windows Session”](#) for details.

17.16.3. How to Make Sun Ray Printers Available to a Windows Session

The Windows session is aware only of the print queues specified in the command line when the Windows connector is started. To update the Windows session with the available print queues on the Sun Ray server, you must restart the Windows connector with the relevant print queues specified on the command line.

Before You Begin

- Make sure the print queues are set up on the Sun Ray server. See [Section 17.16.1, “How to Set Up Print Queues \(Oracle Solaris 10\)”](#) and [Section 17.16.2, “How to Set Up Print Queues \(Oracle Linux\)”](#) for details.
- Printer data is created on the Windows system, so make sure to specify the name of the printer's Windows driver and install it on the Windows system. If you make a printer available without specifying a driver, the Windows connector defaults to a PostScript driver.
- To find the printer driver name on a Windows system, check the Windows Registry key at:

```
MyComputer\HKEY_LOCAL_MACHINE\System\CurrentControlSet\
/Control/Print/Environments/Windows NT x86/Drivers/Version-3
```

All the printer drivers installed on the system are displayed on this list.

Steps

- To specify a printer's Windows driver:

```
% /opt/SUNWuttsc/bin/uttsc -r printer:printer-name="windows-printer-driver-name" hostname.domain
```

where `printer-name` is a valid raw print queue on the Sun Ray server and `windows-printer-driver-name` is the name of the printer exactly as shown on the Windows server. Double quotes are required around the name of the printer.

- To make a printer available without specifying a driver:

```
% /opt/SUNWuttsc/bin/uttsc -r printer:printer-name hostname.domain
```

where `printer-name` is a valid raw print queue on the Sun Ray server.

- To make multiple printers available:

```
% /opt/SUNWuttsc/bin/uttsc -r printer:printer1=driver1,printer2=driver2 hostname.domain
```

17.16.4. How to Manage Printer Configurations for Users

Sun Ray Software automatically keeps track of the printer configuration changes on a remote Windows system for each user. Whenever a user changes the printer configuration on a remote Windows system for any of the printers specified through the `uttsc` command, the Sun Ray server saves those changes to the Sun Ray data store. The Sun Ray server then restores the saved printer configurations whenever the user reconnects to the Windows system through the Windows connector.

The `uttscprinteradm` command helps you manage this information. You can use it to list the available printer information and to perform cleanup in case of user or printer deletion. See the `uttscprinteradm` man page for details.

17.16.5. How to Set Up Follow-Me Printing

This procedure describes how to set up Follow-Me Printing, which retains the user's default printer for a Windows session regardless of the Sun Ray Client being used.



Note

Implementation of Follow-Me Printing relies on technology not available by default and non-public Sun Ray interfaces as well as the use of certain public Sun Ray interfaces for purposes other than their intended use. For these reasons, this feature is not a supported feature.

1. For each Sun Ray Client of interest, specify a printer in the Sun Ray data store.
 - a. Log in to the Admin GUI.
 - b. Click Desktop Units.
 - c. Click the Sun Ray Client for which you want to set up Follow-Me Printing.
 - d. Click Edit and add the printer's name to the Other Information field.
2. Create a shell script that queries the printer name stored in the Sun Ray data store for the current Sun Ray Client and writes that name to the user's `$HOME/.printers` file.

For example:

```
#!/bin/sh
if [ 'uname' = Linux ] ; then
theFlag="-P"
fi
theMACAddress='cd $theFlag $UTDEVROOT ; /bin/pwd | sed 's/. *?.....?/1/'
thePrinter='/opt/SUNWut/sbin/utdesktop -o |
grep $theMACAddress |
/usr/bin/awk -F, '{print $3}'
echo "_default $thePrinter" > $HOME/.printers
```

3. Use `utaction` to invoke the script above on an initial connection and subsequently whenever a user moves from one Sun Ray Client to another.

Provide the following scripts based on the user's login manager:

- `dtlogin` display manager (Oracle Solaris 10) - `xsession.d` script
- `gdm` display manager (Oracle Linux and Oracle Solaris 11) - `xinitrc.d` script

For example, you might create the script `/usr/dt/config/Xsession.d/1100.SUNWut` for `dtlogin` or `/etc/X11/xinit/xinitrc.d/1100.SUNWut` for `gdm` as follows:

```
#!/bin/sh
/opt/SUNWut/bin/utaction -i -c path-to-script &
```

where *path-to-script* is the path to the script you created to retrieve the printer name in step 2.



Note

The name `1100.SUNWut` is chosen purposely in this case to ensure that the script is run or sourced after the existing script `0100.SUNWut`. The `0100.SUNWut` script is responsible for setting `$UTDEVROOT`, which is needed by the first sample script in step 2.



Note

For information about the bundled `gdmgreeter`, see the `kiosk` man page.

4. Modify your Kiosk session script arguments to redirect the printer to Windows.

You may modify these arguments through the Admin GUI. In this example you need to add the argument `-r printer:_default` to the existing arguments, resulting in an argument list similar to the following:

```
-t 1800 -- -m -b -r printer:_default myHost
```

where *myHost* corresponds to the server argument passed to `uttsc`.

17.16.6. Printers Troubleshooting

17.16.6.1. Problem: "Failed to open the printer port" message.

Verify that the printer node used for configuring the printer has been created and is available under `/tmp/SUNWut/units/IEEE802.macid/dev/printers`.

If the printer node is not available, reboot the client.

17.17. Accessing Serial Devices

The Windows connector provides serial device mapping, which enables users to access external serial devices connected to a Sun Ray Client or an Oracle Virtual Desktop Client running on a Windows client computer. When initiating the Windows connector, you need to configure the device mapping through the `-r comport:` option of the `uttsc` command.

Here is an example of mapping a serial device mounted on `$UTDEVROOT/dev/term/a` to the device name `SER_A`.

```
uttsc -r comport:SER_A=$UTDEVROOT/dev/term/a -i ip_addr
```

For details on how to determine where serial devices are mounted for a desktop client, see [Section 15.4, “Accessing Serial Devices and USB Printers”](#).



Note

USB-to-serial adapters are not accessible through the generated device nodes. You must use USB redirection in a Windows session to access a serial device connected through a USB-to-serial adapter.

Once the serial device is mapped, there are various ways to verify that the serial device is available as mapped, such as `SER_A` in the previous example. See [Table 17.11, “Windows Commands to Verify Available Serial Devices”](#) for the list of some recommended commands and the notes that follow.

Table 17.11. Windows Commands to Verify Available Serial Devices

Windows Version	chgport /q	net use	mode	PuTTY
Windows XP	Yes, see notes	No	No	Yes
Windows Server 2003 R2	Yes	No	No	Yes
Windows 7, Windows 8, Windows Server 2008 R2, Windows Server 2012	Yes	No	Yes	Yes

Here are some notes for [Table 17.11, “Windows Commands to Verify Available Serial Devices”](#).

- `chgport /q` - This command is not part of Windows XP. However, you can copy both the `change.exe` and `chgport.exe` executable files from a Windows Server 2003 R2 system to a Windows XP system and use them. Under Windows XP, the `chgport \q` command will display the device if you map it using the following command, where `COMn:` is an unused COM port and `mapped_name` is the name of the mapped device:

```
net use \\COMn:\\tsclient\mapped_name
```

- `net use` and `mode` - You should run both of these commands from the Windows command prompt.
- `PuTTY` - This command is available free at <http://www.chiark.greenend.org.uk/%7Esgtatham/putty/download.html>. Do not type a colon (<literal>:</literal>) when entering a serial port in PuTTY, even if the mapped name contains a colon.

17.18. uttsc Error Messages

[Table 17.12, “uttsc Error Messages”](#) provides the list of `uttsc` error messages.

Table 17.12. uttsc Error Messages

Message	Comments
Error (%d): Unable to establish data store connection.	The Windows connector was unable to open a connection to the Sun Ray data store. Ensure that the Sun Ray data store has been configured for Sun Ray software and is reachable. Also, ensure that the Windows connector has been successfully configured before launching it.
Error(%d): Unable to determine SRSS version.	The Windows connector could not determine the Sun Ray Software version information. Ensure that the correct version is installed and configured successfully.

Message	Comments
Error(%d): Unable to launch Sun Ray Connector. Only SRSS x.x and above are supported.	Ensure that the correct version of Sun Ray Software is installed.
Sun Ray session is not connected, please try again.	Ensure that the Windows connector is being launched from a valid connected Sun Ray session.
Cannot obtain client MAC address.	The Windows connector was unable to contact the Sun Ray Authentication Manager to retrieve the client's MAC address. Ensure that this daemon is reachable.
Error: Sun Ray Token ID cannot be determined. Sun Ray Connector can only be launched from a Sun Ray session.	The Windows connector was launched from a non-Sun Ray session (for example, telnet or console). It can only be launched from a connected client session.
Unable to create new audio device. Using default audio device.	<code>utaudio</code> failed to create a new audio device. Check the messages logged by <code>utaudio</code> for more information. The Windows connector will try to use the default audio device for the session.
Device <code>device_name</code> is not allocated. Audio will not work in this session. Continuing..	On Oracle Solaris Trusted Extensions platforms, if the default audio device is not allocated, then the Windows connector will not be able to use any new audio device or the default audio device. In this case, the Windows connector session will proceed but without audio support.
Warning. Printer preferences will not be stored. Please run <code>utconfig -c</code> to complete configuration before launching Sun Ray Connector.	If <code>utconfig -c</code> has not been run before the Windows connector is launched, the printer preferences as sent by the Windows system will not be stored and hence cannot later be reused. This error is not fatal. The session will continue to be launched.
Unable to connect to Sun Ray Connector Proxy. Please ensure <code>utconfig -c</code> has been run before launching the Windows connector.	Make sure the proxy daemon (<code>uttsrpd</code>) is up and running. If the Windows connector is launched before <code>utconfig -c</code> has been run to configure it, then the Windows connector proxy is not reachable. This message occurs only on Oracle Solaris systems.
Unable to launch Sun Ray Connector. Please ensure <code>utconfig</code> has been run before launching the Sun Ray Connector.	If the Windows connector is launched without having configured Sun Ray data store using <code>utconfig</code> (from Sun Ray Software), then the connector cannot be used.

17.18.1. General Troubleshooting

17.18.1.1. Problem: Unexpected Time Zone Value

`uttsr` only considers time zones listed in `/usr/share/lib/zoneinfo/tab/zone_sun.tab` (for Oracle Solaris) and `/usr/share/zoneinfo/zone.tab` (for Oracle Linux), as valid zones that can be converted into the equivalent time zones in the Windows session. If the time zone is set to a value other than those defined in these files, then the time zone value in the Windows session can be unexpected.

Chapter 18. VMware View Connector

Table of Contents

18.1. VMware View Connector Overview	271
18.2. Requirements	271
18.3. Configuring the VMware View Environment	272
18.3.1. How to Disable Secure Tunnel Connections to View Desktops	272
18.3.2. How to Enable Non-SSL Connections to the View Connection Server	272
18.3.3. How to Enable SSL Connections to the View Connection Server	272
18.4. Configuring the VMware View Connector Kiosk Session Type	273
18.4.1. How to Configure a Kiosk Mode Session Type for the VMware View Connector	273
18.5. VMware View Connector Troubleshooting	274
18.5.1. Error Messages	274
18.5.2. Desktop tries to open, but immediately disconnects	275

This chapter describes how to configure the VMware View connector, so desktop users can connect to their Windows virtual desktops provided through the VMware View Manager.

18.1. VMware View Connector Overview

Sun Ray Software provides a VMware View connector, which is an RDP client that enables desktop client users to connect to their Windows virtual desktop provided through the VMware View Manager. The VMware View connector is an extension of the Windows connector. Once the VMware View connector is configured through kiosk mode, users can log in to their View desktop using their Active Directory user name and password. Users do not have to reenter their password at the Windows login screen.

The Sun Ray Clients are a certified VMware View and VMware Virtual Desktop Manager (VDM) client solution. They are listed in VMware's Hardware Compatibility Guide at <http://www.vmware.com/resources/compatibility>.

For details on how to install and configure the VMware View product, refer to the [VMware View documentation](#).

18.2. Requirements

The following VMware View releases are supported by Sun Ray Software 5.4 at initial release:

- VMware View 5.1
- VMware View 5.0
- VMware View 4.6
- VMware View 4.5
- VMware View 4.0
- VMware View 3.1

For the latest list of supported VMware View releases, refer to VMware's Hardware Compatibility Guide at <http://www.vmware.com/resources/compatibility>.

For the list of supported Windows remote desktops, see [Section 3.1.3, "Windows Remote Desktop Support"](#).

18.3. Configuring the VMware View Environment

This section describes the changes needed to the VMware View environment before configuring and using the VMware View connector. It is assumed that the VMware View environment is properly configured and working before using the VMware View connector. Refer to the [VMware View documentation](#) for details.

18.3.1. How to Disable Secure Tunnel Connections to View Desktops

The VMware View connector does not support tunneled connections when accessing View desktops. The following steps describe how to disable secure tunnel connections to View desktops through a View Connection Server.

This procedure is based on the VMware View 5.0 release.

1. Log in to the VMware View Administrator.
2. From the **View Configuration** menu in the left-hand navigation pane, click **Servers**.
3. From the Servers panel, select the appropriate View Connection Server and click **Edit**.
4. From the Edit View Connection Server Settings pop-up screen, deselect **Use Secure Tunnel connection to desktop**.
5. Click **OK**.

18.3.2. How to Enable Non-SSL Connections to the View Connection Server

If (Secure Sockets Layer) SSL is not required, use the following steps to enable non-SSL connections to the View Connection Server.

**Note**

This configuration is not possible in VMware View 5.1 or later, because only SSL connections are supported.

This procedure is based on the VMware View 5.0 release.

1. Log in to the VMware View Administrator.
2. From the **View Configuration** menu in the left-hand navigation pane, click **Global Settings**.
3. Click **Edit**.
4. Deselect **Require SSL for client connections and View Administrator** in the pop-up screen.
5. Click **OK**.

18.3.3. How to Enable SSL Connections to the View Connection Server

The default SSL certificate created from the VMware View Manager installation must be imported into the Sun Ray server to enable SSL connections to desktops. The following steps assume that you have generated and stored a new certificate on the View Connection Server.

1. Export the certificate from the keystore on the View Connection Server:

```
# keytool -export -keystore keys.p12 -storetype pkcs12 -file vmware.cer
```

2. Copy the `vmware.cer` file to the Sun Ray server.

3. Import the certificate into a keystore on the Sun Ray server:

```
# keytool -import -file vmware.cer -trustcacerts -v -keystore \
/etc/opt/SUNWkio/sessions/vdm/keystore
```

4. Edit the kiosk script (`/etc/opt/SUNWkio/sessions/vdm/vdm`) and modify the line that begins with `javaKeyStorePass=` to include the password for the keystore.
5. Restart the Sun Ray server via the Admin GUI.



Note

The administrator may choose to import the certificate into the default keystore of the server's Java installation instead of following steps 3 and 4. If this is done, the kiosk script must be modified and all references to `javaKeyStore` and `javaKeyStorePass` should be removed.

18.4. Configuring the VMware View Connector Kiosk Session Type

The following steps describe how to configure the VMware View connector, which is provided as a pre-defined kiosk session type.

To set up the VMware View environment for Sun Ray Software, see [Section 18.3, “Configuring the VMware View Environment”](#).

For more information about configuring kiosk mode, see [Chapter 10, Kiosk Mode](#).

18.4.1. How to Configure a Kiosk Mode Session Type for the VMware View Connector

1. Log in to the Admin GUI.
2. Click the Advanced tab and Kiosk Mode sub-tab. Then click **Edit**.
3. Choose **VMware View Manager Session** from the **Session** (Session Type) menu.
4. Modify the session parameters.
5. Add session arguments to the Arguments field at the bottom in the format:

```
[session-type-arguments] [-- uttsc-arguments]
```

See [Table 18.1, “Kiosk Session Arguments for VMware View Connector”](#) for the list of valid session arguments.

Table 18.1. Kiosk Session Arguments for VMware View Connector

Argument	Description
<code>-s server</code>	VMware View Connection Server hostname.
<code>-https</code>	Use SSL connection to VMware View Connection Server (default).
<code>-http</code>	Do not use SSL connection to VMware View Connection Server.
<code>-p port-number</code>	VMware View Connection Server port number.
<code>-t seconds</code>	If no smart card is used, the length of inactivity before the user is automatically logged out of the desktop selection dialog. Default value is 3 minutes.

Argument	Description
<code>-no-auto-login</code>	Users are automatically forwarded to their desktop if there is only one desktop. This flag disables this behavior.
<code>-d domain</code>	This domain name will be preselected in the login screen, if available.
<code>-- uttsc-arguments</code>	Specify any valid <code>uttsc</code> arguments. For detailed information on these options, refer to the <code>uttsc</code> man page.

**Note**

The Sun Ray Software is capable of supporting Windows Network Level Authentication (NLA), but VMware View does not support NLA on non-Windows based View clients. You must use the standard RDP authentication with VMware View. To configure RDP authentication, enable RDP authentication on the guest OS and add the `-N off` option to the `uttsc` arguments field.

6. Configure the server to use kiosk mode for card and non-card users.
 - a. Click the System Policy sub-tab on the Advanced menu.
 - b. Enable **Kiosk Mode** for both card and non-card users.
7. Click **Save**.

All new or restarted sessions matching the policy configuration to use kiosk mode will access the new session type.

18.5. VMware View Connector Troubleshooting

This section provides troubleshooting tips for the VMware View connector.

18.5.1. Error Messages

In the event the software does not work as expected, look at the log messages located in the Sun Ray server's `/var/opt/SUNWut/log/messages` file. Error messages related to the VMware View connector begin with `kiosk:vdm`. There may also be useful information in the `/var/dt/Xerrors` file.

18.5.1.1. Error connecting to VDM

server:javax.net.ssl.SSLException:java.lang.RuntimeException:Unexpected error:java.security.InvalidAlgorithmParameterException:the trustAnchors parameter must be non-empty

The SSL certificate is not set up correctly on the Sun Ray server.

Solution: See [Section 18.3.3, "How to Enable SSL Connections to the View Connection Server"](#).

18.5.1.2. This desktop is currently not available. Please try connecting to this desktop again later, or contact your system administrator. The desktop sources for this desktop are not responding. Please try connecting to the desktop again later, or contact your system administrator

The desktop is not set up properly or is already in use. For example:

- Someone is logged into the machine (over remote desktop or via the console in VMware vCenter).
- The machine is powering on/off, or suspending.
- No free desktops exist for that user.
- VMware View Agent is not installed on the desktop, or it is not working correctly. Check that the desktop status is available in the VMware View Connection Server.
- Active Directory and/or DNS is not set up properly on the desktop.
- There is a network communication problem between VMware View Connection Server and the desktop.
- A Windows firewall is blocking connections to the desktop.

18.5.1.3. Exception in thread "main" java.lang.NoClassDefFoundError

An incorrect version of Java software is in use.

Solution: Install Java version 1.5 or 1.6.

18.5.1.4. Connection tunneling is required to connect to the desktop, but it is not supported by this client

Connection tunneling is not supported by the VMware View connector.

Solution: See [Section 18.3.1, "How to Disable Secure Tunnel Connections to View Desktops"](#).

18.5.1.5. Error connecting to VMware View Manager: java.io.FileNotFoundException: http://ip_address//broker/xml: 404: Not Found

The VMware View connector is trying to use HTTP to connect to the View desktop, and non-SSL connections are not configured or not supported.

Solution: For VMware View 5.1 or later, HTTP is not supported. So, you must configure the kiosk session type to use HTTPS (which is the default) and make sure your VMware View environment is properly configured as described in [Section 18.3.3, "How to Enable SSL Connections to the View Connection Server"](#). If you are using VMware View 5.0 or earlier and you want to use HTTP, you must configure non-SSL connections as described in [Section 18.3.2, "How to Enable Non-SSL Connections to the View Connection Server"](#).

18.5.2. Desktop tries to open, but immediately disconnects

Solution: Try to connect to the desktop manually from the Sun Ray server using the `/opt/SUNWuttsc/bin/uttsc desktop-IP` command. A remote desktop connection to the virtual machine should open. If it fails, it can provide an error message with further information.

Chapter 19. Alternate Network Configurations

Table of Contents

19.1. Alternate Network Configurations Overview	277
19.2. Updating the Default <code>/etc/hosts</code> File Before Configuring Sun Ray Network (Oracle Linux)	278
19.3. Using a Shared Network Configuration Without External DHCP Services	278
19.3.1. Shared Network Configuration Worksheet	278
19.3.2. How to Configure a Sun Ray Server on a Shared Network to Provide DHCP Services	279
19.3.3. How to List the Current Network Configuration	280
19.3.4. How to Delete a LAN Subnet	280
19.3.5. Example Shared Network Setups	281
19.4. Using a Private Network Configuration	289
19.4.1. Private Network Configuration Worksheet	290
19.4.2. How to Configure a Sun Ray Server in a Private Network	292
19.4.3. How to List the Current Network Configuration	293
19.4.4. How to Print a Private Network Configuration	293
19.4.5. How to Delete an Interface	293
19.4.6. Example Private Network Setup	293
19.5. Sun Ray Client Initialization Requirements Using DHCP	296
19.5.1. DHCP Basics	296
19.5.2. DHCP Parameter Discovery	297
19.5.3. DHCP Relay Agent	298
19.5.4. Simplifying DHCP Configuration of Remote Sun Ray Clients	298
19.5.5. Standard DHCP Parameters	299
19.5.6. Vendor-specific DHCP Options	300
19.5.7. Encapsulated Options	302
19.6. Failover Groups	303
19.6.1. Network Topologies	303
19.6.2. Setting Up IP Addressing	305
19.6.3. Sun Ray Server Failover Group Worksheet	309

This chapter provides alternative network configurations that are supported for Sun Ray Software.

19.1. Alternate Network Configurations Overview

In the alternate network configurations, the Sun Ray server is also used as a DHCP server. Using a shared network configuration with external DHCP services is the recommended way to set up a Sun Ray environment, which is described in [Chapter 2, Planning a Sun Ray Network Environment](#).

The `utadm` command is used to manage the Sun Ray network interfaces. Note the following information:

- If the IP addresses and DHCP configuration data are not set up properly when the interfaces are configured, then the failover feature will not work as expected. In particular, configuring the Sun Ray server's interconnect IP address as a duplicate of any other server's interconnect IP address may cause the Sun Ray Authentication Manager to generate "Out of Memory" errors.
- If you make manual changes to your DHCP configuration, you will have to make them again whenever you run `utadm` or `utfwadm`.
- (Oracle Solaris only) If you press `CTRL-C` while performing `utadm` configuration, `utadm` might not function correctly the next time it is invoked. To correct this condition, type `dhtadm -R`.

19.2. Updating the Default `/etc/hosts` File Before Configuring Sun Ray Network (Oracle Linux)

The `utadm` command is not able to parse the default `/etc/hosts` file when trying to configure the Sun Ray network. The following error may occur when using the `utadm -a` or `utadm -A` commands on an Oracle Linux-based Sun Ray server:

```
Error: host IP address must be set.
Set host IP address and try again
```

On Oracle Linux systems, you must make sure that the system's host name specified during the Oracle Linux installation is configured properly in the `/etc/hosts` file. The system's host name must be on a separate line in the `/etc/hosts` file with an IP address matching the host's primary IP address.

Here is an example of an `etc/hosts` file that contains the system's host name, `srshost`, on the same line as `localhost`:

```
127.0.0.1      srshost localhost.localdomain localhost
::1           localhost6.localdomain6 localhost6
```

The `srshost` host name must be on its own line for the `utadm` command to work:

```
127.0.0.1      localhost.localdomain localhost
::1           localhost6.localdomain6 localhost6
192.168.1.1    srshost
```

19.3. Using a Shared Network Configuration Without External DHCP Services

If you have a shared network configuration where DHCP services are not available, the Sun Ray server can be configured to provide those services. This configuration adds more complexity to the initial and ongoing Sun Ray network administration. See [Chapter 2, *Planning a Sun Ray Network Environment*](#) for overall information about using shared networks with available DHCP services.

You can use the `utadm -A subnet` command to configure a Sun Ray server to provide DHCP services. The Sun Ray server can be configured to respond with network/IP information, device configuration information, or both.

19.3.1. Shared Network Configuration Worksheet

Fill out [Table 19.1, “Shared Network Configuration Worksheet”](#), so that the information is readily available during the actual configuration process. This worksheet is for configuring a Sun Ray server in a shared network without an external DHCP service.

- Values that are provided in *italics* are only examples and should *not* be used.
- Values provided in normal font are defaults and can be used.
- Superscripted numbers ^(#) refer to footnotes at the end of each section.



Note

The blank rows in the worksheets are provided for you to add additional information about your environment if you choose to print the worksheets.

Table 19.1. Shared Network Configuration Worksheet

Aspect or Variable	Default Value, Example, or (Other)	Your Primary Server Value	Your Secondary Server Value
Configuring the Sun Ray interconnect interface (Provide the start time) using <code>utadm</code>			
• Subnetwork	192.168.128.0		
• Host address ⁽¹⁾	192.168.128.1		
• Net mask	255.255.255.0		
• Net address	192.168.128.0		
• Host name ⁽¹⁾	<code>hostname-interface-name</code>		
If the Sun Ray server is used for IP address allocation:			
• First Sun Ray Client address ⁽²⁾	192.168.128.16		
• Number of Sun Ray Client addresses ⁽²⁾	<code>X</code>		
• Firmware server ⁽³⁾	192.168.128.1		
• Router ⁽³⁾	192.168.128.1		
Specify additional server list? (optional)	(yes or no)		
• If yes, filename	<code>filename</code>		
• Or, Server IP address	192.168.128.2		

⁽¹⁾ These values are different for each Sun Ray server, even if that server is part of a failover group.

⁽²⁾ These values must be unique among the servers in a failover group. The following guidelines can help you determine what addresses to allocate for each Sun Ray server:

- $X = (\text{Number of clients} / (\text{Number of servers} - 1)) - 1$.
- First unit address for primary server = 192.168.128.16.
- Last unit address for all servers = $X + \text{first unit address}$. If last unit address is greater than 240, reduce to 240.
- First unit address for secondary servers = 1 + last unit address of previous server. If first unit address is greater than 239, configure for a class B network. Example: 120 clients, 4 servers. $X = 39$.

⁽³⁾ These values are the same as the interface host address by default.

19.3.2. How to Configure a Sun Ray Server on a Shared Network to Provide DHCP Services

This procedure shows how to configure a Sun Ray Server to provide DHCP services to Sun Ray Clients.

1. Log in as the superuser of the Sun Ray server.
2. Configure the Sun Ray LAN subnet:

```
# /opt/SUNWut/sbin/utadm -A subnet#
```

where `subnet#` is the identifying number of the subnet, such as 192.168.128.0.

The `utadm` script begins configuring DHCP for the Sun Ray interconnect, restarts the DHCP daemon, and configures the interface. The script then lists the default values and asks whether they are acceptable.

**Note**

If the IP addresses and DHCP configuration data are not set up correctly when the interfaces are configured, the failover feature cannot work properly. In particular, configuring the Sun Ray server's subnet IP address as a duplicate of any other server's subnet IP address may cause the Sun Ray Authentication Manager to issue Out of Memory errors.

3. Evaluate the default values.

- If you are satisfied with the default values and the server is not part of a failover group, answer y.
- Otherwise, answer n and accept whatever default values are shown by pressing Return or provide the correct values from the worksheet.

The `utadm` script prompts for the following:

- New netmask (255.255.255.0)
- New first Sun Ray Client address (192.168.128.16)
- Total number of Sun Ray Client addresses
- New authorization server address (192.168.128.1)
- New firmware server address (192.168.128.10)
- New router address (192.168.128.1)
- An additional server list.

If you answer yes, it requests either a file name (`_filename_`) or a server IP address (192.168.128.2)

4. The `utadm` script again lists the configuration values and asks whether they are acceptable.

- If not, answer n and revise the answers you provided in Step 3.
- If the values are correct, answer y. The `utadm` script configures the Sun Ray Client firmware versions and restarts the DHCP daemon.

5. Repeat this procedure for each of the secondary servers in your failover group.

6. If a router is between the Sun Ray server and the Clients, configure bootp forwarding in the routers.

19.3.3. How to List the Current Network Configuration

```
# utadm -l
```

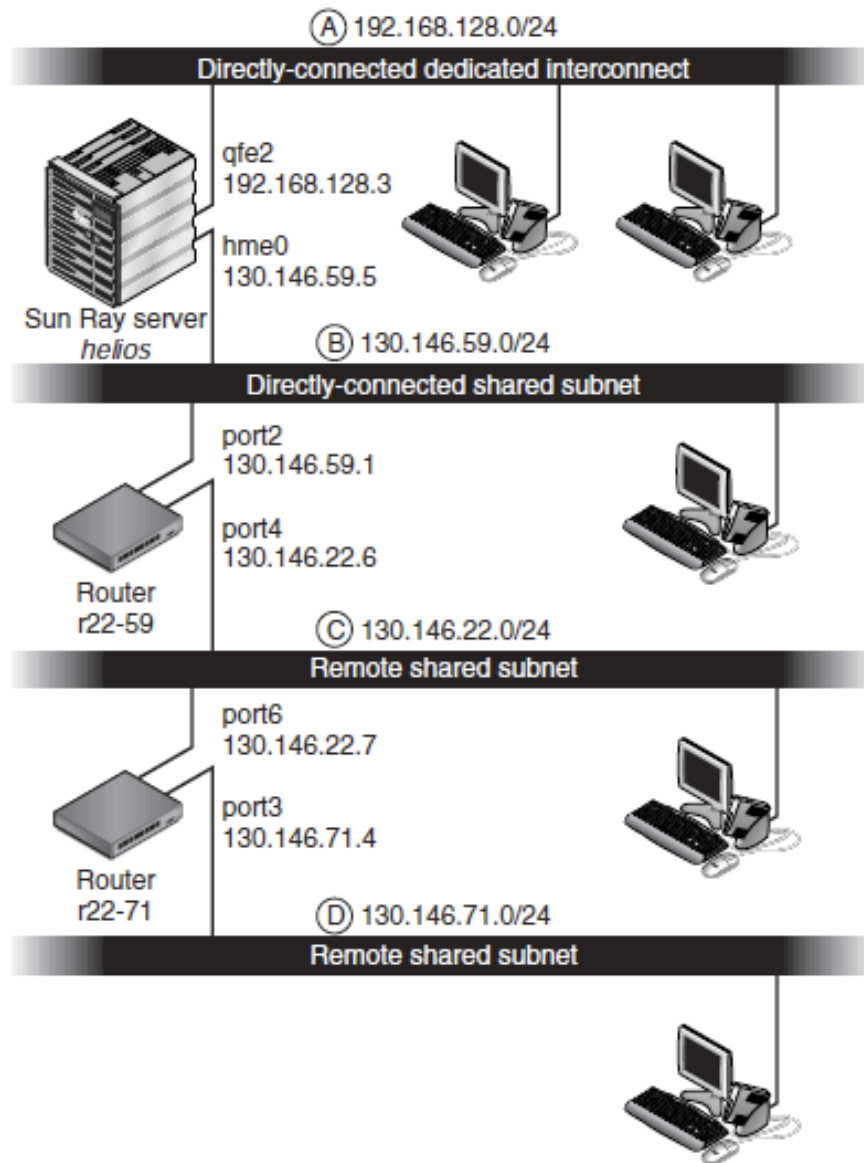
19.3.4. How to Delete a LAN Subnet

```
# utadm -D subnet#
```

19.3.5. Example Shared Network Setups

The following section presents an example of a Sun Ray Client deployment on shared networks B, C, and D as shown in [Figure 19.1, “Example of Alternate Shared Network Topology”](#).

Figure 19.1. Example of Alternate Shared Network Topology



19.3.5.1. Deployment on a Directly Connected Shared Subnet

Subnet B in [Figure 19.1, “Example of Alternate Shared Network Topology”](#) is a directly connected shared subnet that uses IP addresses in the range `130.146.59.0/24`. The Sun Ray server `helios` is attached to the interconnect through its `hme0` network interface, which has been assigned the IP address `130.146.59.5`. The answers to the three predeployment questions are as follows:

- From which DHCP server will Clients on this subnet get their basic IP networking parameters?

In a shared subnet scenario, you must choose whether a DHCP service on the Sun Ray server or some external DHCP service will provide the client with basic network parameters. If the enterprise already has a DHCP infrastructure that covers this subnet, it probably supplies basic network parameters. If no such infrastructure exists, configure the Sun Ray server to provide basic network parameters.

- **From which DHCP server will clients on this subnet get additional configuration parameters to support features such as firmware download?**

The administrator must choose whether to supply additional configuration parameters to the client and, if so, whether to use a DHCP service on the Sun Ray server or some external DHCP service for this purpose. On a directly connected shared subnet, it is possible to deploy clients without providing additional parameters at all, but this configuration is not desirable because it deprives the client of a number of features, including the ability to download new firmware.

Administrators of an already established DHCP infrastructure might be unable or unwilling to reconfigure that infrastructure to provide additional Sun Ray configuration parameters, so having the Sun Ray server provide these parameters is usually more convenient. This setup can be desirable even when the established infrastructure is capable of delivering the additional parameters. This setup enables SRSS commands to be used to manage the values of the additional configuration parameters when those values need to be changed in response to software upgrades or patch installations on the Sun Ray server.

For instance, a patch that delivers new client firmware could automatically update the firmware version string that is delivered to the client. However, if the firmware version parameter is supplied by some external DHCP service, an administrator must manually edit the firmware version parameter string in the external DHCP configuration rules to reflect the new firmware version delivered by the patch. This activity is time-consuming and error-prone, as well as unnecessary.

- **How will clients on this subnet locate their Sun Ray server?**

Use one of the optional additional configuration parameters to report the location of the Sun Ray server to the client. If additional configuration parameters are not supplied to the client at all, the client has no indication of the location of any Sun Ray server. In these circumstances, the client attempts to discover the location of a Sun Ray server by using a broadcast-based mechanism. However, the clients broadcast packets propagate only on the local subnet so, in the case of a remote subnet, the broadcast cannot reach the Sun Ray server, and contact cannot be established.

The following examples illustrate two configurations of the directly connected shared subnet. In the first example, the Sun Ray server delivers both basic networking parameters and additional parameters. In the second example, an external DHCP service supplies basic networking parameters but no additional parameters are provided to the client, which must establish contact with the Sun Ray server through its local subnet broadcast discovery mechanism.

The most likely case, where an external DHCP service provides basic networking parameter and the Sun Ray server provides additional parameters, is illustrated by an example in [Section 19.3.5.2, “Deployment on a Remote Subnet”](#).

19.3.5.1.1. Directly Connected Shared Subnet: Example 1

In this example, the answers to the three predeployment questions are as follows:

- **From which DHCP server will clients on this subnet get their basic IP networking parameters?**

From the Sun Ray server.

- **From which DHCP server will clients on this subnet get additional configuration parameters to support features such as firmware download?**

From the Sun Ray server.

- **How will clients on this subnet locate their Sun Ray server?**

The clients will be informed of the location of the Sun Ray server through an additional configuration parameter delivered when Sun Ray services are restarted.

1. Configure the Sun Ray server to provide both basic and additional parameters to the shared subnet.

DHCP service for clients on a shared subnet is configured through the `utadm -A subnet` command. In this example, the shared subnet has network number `130.146.59.0`, so the appropriate command is `utadm -A 130.146.59.0`.

```
# /opt/SUNWut/sbin/utadm -A 130.146.59.0
Selected values for subnetwork "130.146.59.0"
net mask: 255.255.255.0
no IP addresses offered
auth server list: 130.146.59.5
firmware server: 130.146.59.5
router: 130.146.59.1
Accept as is? ([Y]/N): n
netmask: 255.255.255.0 (cannot be changed - system defined netmask)
Do you want to offer IP addresses for this subnet? (Y/[N]): y
new first Sun Ray address: [130.146.59.4] 130.146.59.200
number of Sun Ray addresses to allocate: [55] 20
new auth server list: [130.146.59.5]
To read auth server list from file, enter file name:
Auth server IP address (enter <CR> to end list):
If no server in the auth server list responds, should an auth server be located by
broadcasting on the network? ([Y]/N):
new firmware server: [130.146.59.5]
new router: [130.146.59.1]
Selected values for subnetwork "130.146.59.0"
net mask: 255.255.255.0
first unit address: 130.146.59.200
last unit address: 130.146.59.219
auth server: 130.146.59.5
firmware server: 130.146.59.5
router: 130.146.59.1
auth server list: 130.146.59.5
Accept as is? ([Y]/N):
### Building network tables - this will take a few minutes
### Configuring firmware version for Sun Ray
All the units served by "helios" on the 130.146.59.0
network interface, running firmware other than version
"2.0_37.b,REV=2002.12.19.07.46" will be upgraded at
their next power-on.
### Configuring Sun Ray Logging Functions
### stopped DHCP daemon
### started DHCP daemon
#
```

The default values initially suggested by `utadm` were not appropriate. Specifically, this server would not have offered any IP addresses on the `130.146.59.0` subnet because `utadm` assumes that basic networking parameters, including IP addresses, are provided by some external DHCP service when the client is located on a shared subnet. In this example, however, the Sun Ray server is required to provide IP addresses, so the administrator replied `n` to the first "Accept as is?" prompt and was given the opportunity to provide alternative values for the various parameters. Twenty IP addresses, starting at `130.146.59.200`, were made available for allocation to DHCP clients on this subnet.

- Restart Sun Ray services on the Sun Ray server by issuing the `utstart` command to fully activate Sun Ray services on the shared subnet.

```
# /opt/SUNWut/sbin/utstart
A warm restart has been initiated... messages will be logged to /var/opt/SUNWut/log/messages.
```

19.3.5.1.2. Directly Connected Shared Subnet: Example 2

In this example, the answers to the three predeployment questions are as follows:

- From which DHCP server will clients on this subnet get their basic IP networking parameters?**

From an external DHCP service.

- From which DHCP server will clients on this subnet get additional configuration parameters to support features such as firmware download?**

The clients will not be supplied with additional parameters.

- How will clients on this subnet locate their Sun Ray server?**

By using the local subnet broadcast discovery mechanism.

In this example, the Sun Ray server does not participate in client initialization at all. Configuration steps are still required on the Sun Ray server because it responds by default only to clients located on directly connected dedicated interconnects. It responds to clients on shared subnets only if the `utadm -L` on command has been executed. Running the `utadm -A subnet` command to activate DHCP on the Sun Ray server for a shared subnet, as in this example, implicitly executes `utadm -L` on. If `utadm -A subnet` has not been run, the administrator must run `utadm -L` on manually to enable the server to offer sessions to clients on the shared subnet.

- Configure the external DHCP service.

Determining how to configure the external DHCP infrastructure to provide basic networking parameters to the clients on this subnet is beyond the scope of this document. Note the following guidelines:

- If the external DHCP service does not have its own direct connection to this subnet, the administrator must configure a DHCP Relay Agent to deliver DHCP traffic on this subnet to the external DHCP service. The most likely location for such a Relay Agent would be on a router in this subnet, in this case, the router named `r22-59` in [Figure 19.1, “Example of Alternate Shared Network Topology”](#). For a brief introduction to this topic, refer to [Section 19.5, “Sun Ray Client Initialization Requirements Using DHCP”](#).
- An existing external DHCP service might need to have its IP address allocation for this subnet increased in order to support the new clients. This requirement applies whenever additional DHCP clients are placed on a subnet. You might also want to reduce the lease time of addresses on this subnet so that addresses become eligible for reuse quickly.

- Configure the Sun Ray server to accept client connections from shared subnets by running the following command:

```
# /opt/SUNWut/sbin/utadm -L on
### Turning on Sun Ray LAN connection
NOTE: utstart must be run before LAN connections will be allowed
```

- Restart Sun Ray services on the Sun Ray server by issuing the `utstart` command to fully activate Sun Ray services on the shared subnet.

```
# /opt/SUNWut/sbin/utstart
```

```
A warm restart has been initiated... messages will be logged to /var/opt/SUNWut/log/messages.
```

19.3.5.2. Deployment on a Remote Subnet

Subnets C and D in [Figure 19.1, “Example of Alternate Shared Network Topology”](#) are remote shared subnets.

Subnet C uses IP addresses in the range `130.146.22.0/24`. Subnet D uses IP addresses in the range `130.146.71.0/24`. The Sun Ray server named `helios` has no direct attachment to either of these subnets. This characteristic defines them as remote. The answers to the three predeployment questions are as follows:

- **From which DHCP server will clients on this subnet get their basic IP networking parameters?**

In a shared subnet scenario, the administrator must choose whether a DHCP service on the Sun Ray server or some external DHCP service will provide the client with basic network parameters.

If the enterprise already has a DHCP infrastructure that covers this subnet, it probably supplies basic network parameters. If no such infrastructure exists, configure the Sun Ray server to provide basic network parameters.

- **From which DHCP server will clients on this subnet get additional configuration parameters to support features such as firmware download?**

The administrator must choose whether additional configuration parameters will be supplied to the client, and, if so, whether they will be supplied by a DHCP service on the Sun Ray server or by some external DHCP service.

Administrators of an established DHCP infrastructure might be unable or unwilling to reconfigure it to provide additional Sun Ray configuration parameters, so having the Sun Ray server provide them is usually more convenient. This setup can be desirable even when the established infrastructure is capable of delivering the additional parameters. This setup enables you to use Sun Ray Software commands to manage the values of the additional configuration parameters, when those values need to be changed in response to software upgrades or patch installations on the Sun Ray server.

For instance, a patch that delivers new client firmware could automatically update the firmware version string delivered to the client. However, if the firmware version parameter is supplied by some external DHCP service, an administrator must manually edit the firmware version parameter string in the external DHCP configuration rules to reflect the new firmware version delivered by the patch. This kind of activity is time-consuming and error-prone as well as unnecessary.

- **How will clients on this subnet locate their Sun Ray server?**

Use one of the optional additional configuration parameters to report the location of the Sun Ray server to the client. If additional configuration parameters are not supplied to the client at all, the client cannot locate a Sun Ray server, so it tries to discover the location of a Sun Ray server by using a broadcast-based mechanism. However, the clients broadcast packets propagate only on the local subnet so they cannot reach a Sun Ray server located on a remote subnet, and cannot establish contact.

The next two examples illustrate representative remote shared subnet configurations. In the first example, an external DHCP service provides basic networking parameters, and the Sun Ray server provides additional parameters. This configuration is by far the most likely for a Sun Ray deployment in an enterprise that has an established DHCP infrastructure.

In the second example, basic networking parameters and a bare minimum of additional parameters, just enough to enable the client to contact a Sun Ray server, are supplied by an external DHCP. In this case, the DHCP service is in a Cisco router. This scenario is less than ideal.

No firmware parameters are delivered to the client, so it cannot download new firmware. The administrator must make some other arrangement to provide the client with new firmware, for instance, by rotating it off this subnet periodically onto an interconnect or onto some other shared subnet where a full set of additional configuration parameters is offered.

19.3.5.2.1. Remote Shared Subnet: Example 1

In this example, in which clients are deployed on subnet C in [Figure 19.1, “Example of Alternate Shared Network Topology”](#), the answers to the three predeployment questions are as follows:

- **From which DHCP server will clients on this subnet get their basic IP networking parameters?**

From an external DHCP service.

- **From which DHCP server will clients on this subnet get additional configuration parameters to support features such as firmware download?**

From the Sun Ray server.

- **How will clients on this subnet locate their Sun Ray server?**

The clients will be informed of the location of the Sun Ray server through an additional configuration parameter delivered once Sun Ray services are restarted. Use the `utadm -A subnet` command as follows to configure DHCP service for clients on a shared subnet.

1. Configure the external DHCP service.

Determining how to configure the external DHCP infrastructure to provide basic networking parameters to the clients on this subnet is beyond the scope of this document. Note the following guidelines:

- If the external DHCP service does not have its own direct connection to this subnet, the administrator must configure a DHCP Relay Agent to deliver DHCP traffic on this subnet to the external DHCP service. The most likely location for such a Relay Agent would be on a router in this subnet, in this case, the router named `r22-59` in [Figure 19.1, “Example of Alternate Shared Network Topology”](#). For a brief introduction to this topic, refer to [Section 19.5, “Sun Ray Client Initialization Requirements Using DHCP”](#).
- An existing external DHCP service might need to have its IP address allocation increased for this subnet to support the new clients. This requirement applies whenever additional DHCP clients are placed on a subnet. You might also want to reduce the lease time of addresses on this subnet so that addresses become eligible for re-use quickly.

2. Arrange to deliver DHCP traffic to the Sun Ray server.

Because the Sun Ray server does not have its own direct connection to this subnet, the administrator must configure a DHCP Relay Agent to deliver the subnet's DHCP traffic to the Sun Ray server. The most likely location for such a Relay Agent would be on a router in this subnet, in this case, the router named `r22-59` in [Figure 19.1, “Example of Alternate Shared Network Topology”](#). For a brief introduction to this topic, refer to [Section 19.5, “Sun Ray Client Initialization Requirements Using DHCP”](#).

- If `r22-59` is running the Cisco IOS, the `ip helper-address` command can be used to activate its DHCP Relay Agent to relay DHCP broadcasts from its 10/100 Ethernet port number 4 to the Sun Ray server at `130.146.59.5`

```
r22-59> interface fastethernet 4
r22-59> ip helper-address 130.146.59.5
```

```
r22-59>
```

- If the external DHCP service also lacks a connection to this subnet, configure a DHCP Relay Agent to forward requests from the client to the following services:
 - The external DHCP service so that the client can obtain basic networking parameters
 - The DHCP service on the Sun Ray server so that the client can obtain additional parameters

The Cisco IOS `ip helper-address` command accepts multiple relay destination addresses, so if, for example, the external DHCP service could be contacted at `130.146.59.2` on subnet B in [Figure 19.1, “Example of Alternate Shared Network Topology”](#), the appropriate sequence would be:

```
r22-59> interface fastethernet 4
r22-59> ip helper-address 130.146.59.2 130.146.59.5
r22-59>
```



Note

Details of the IOS interaction vary according to the specific release of IOS, the model of the router, and the hardware installed in the router.

3. Configure the Sun Ray server to provide additional parameters to the shared subnet.

Use the `utadm -A subnet` command to configure DHCP service for clients on a shared subnet. In this example, the shared subnet has network number `130.146.22.0`, so the appropriate command is `utadm -A 130.146.22.0`.

```
# /opt/SUNWut/sbin/utadm -A 130.146.22.0
Selected values for subnetwork "130.146.22.0"
net mask: 255.255.255.0
no IP addresses offered
auth server list: 130.146.59.5
firmware server: 130.146.59.5
router: 130.146.22.1
Accept as is? ([Y]/N): n
new netmask:[255.255.255.0]
Do you want to offer IP addresses for this subnet? (Y/[N]):
new auth server list: [130.146.59.5]
To read auth server list from file, enter file name:
Auth server IP address (enter <CR> to end list):
If no server in the auth server list responds, should an auth server be located by
broadcasting on the network? ([Y]/N):
new firmware server: [130.146.59.5]
new router: [130.146.22.1] 130.146.22.6
Selected values for subnetwork "130.146.59.0"
net mask: 255.255.255.0
no IP addresses offered
auth server list: 130.146.59.5
firmware server: 130.146.59.5
router: 130.146.22.6
Accept as is? ([Y]/N):
### Building network tables - this will take a few minutes
### Configuring firmware version for Sun Ray
All the units served by "helios" on the 130.146.22.0
network interface, running firmware other than version
"2.0_37.b,REV=2002.12.19.07.46" will be upgraded at their
next power-on.
### Configuring Sun Ray Logging Functions
### stopped DHCP daemon
### started DHCP daemon
#
```

In this example, the default values initially suggested by `utadm` were not appropriate. Specifically, the default router address to be used by clients on this subnet was not correct because `utadm` guesses that the address of the default router for any shared subnet will have a host part equal to 1. This was a great guess for the directly connected subnet B in [Figure 19.1, “Example of Alternate Shared Network Topology”](#), but it is not correct for subnet C.

The appropriate router address for clients on this subnet is `130.146.22.6` (port 4 of router `r22-59`), so the administrator replied `n` to the first `Accept as is?` prompt and was given the opportunity to provide alternative values for the various parameters.

- Restart Sun Ray services on the Sun Ray server by issuing the `utstart` command to fully activate Sun Ray services on the shared subnet.

```
# /opt/SUNWut/sbin/utstart
A warm restart has been initiated... messages will be logged to /var/opt/SUNWut/log/messages.
```

19.3.5.2.2. Remote Shared Subnet: Example 2

In this example, deploying clients on subnet D in [Figure 19.1, “Example of Alternate Shared Network Topology”](#), the answers to the three predeployment questions are as follows:

- From which DHCP server will clients on this subnet get their basic IP networking parameters?**

From an external DHCP service.

- From which DHCP server will clients on this subnet get additional configuration parameters to support features such as firmware download?**

The clients will not be supplied with the additional parameters required to support firmware download or to activate other advanced client features.

- How will clients on this subnet locate their Sun Ray server?**

The external DHCP service will supply a single additional parameter to inform the client of the location of a Sun Ray server.

In this example, the Sun Ray server does not participate in client initialization at all. Configuration steps are still required on the Sun Ray server because it responds by default only to clients located on directly connected dedicated interconnects. It responds to clients on shared subnets only if the `utadm -L` command has been executed. Running the `utadm -A subnet` command to activate DHCP on the Sun Ray server for a shared subnet, as in this example, implicitly executes `utadm -L` on. If `utadm -A subnet` has not been run, the administrator must run `utadm -L` manually to enable the server to offer sessions to clients on the shared subnet.

- Configure the external DHCP service.

Determining how to configure the external DHCP infrastructure to provide basic networking parameters to the clients on this subnet is beyond the scope of this document. However, for this example, assume that DHCP service is provided by Cisco IOS-based router `r22-71` in [Figure 19.1, “Example of Alternate Shared Network Topology”](#), attached to the `130.146.71.0` subnet through its 10/100 Ethernet port 3. This router can be configured to provide basic networking parameters and the location of a Sun Ray server as follows:

```
r22-71> interface fastethernet 3
r22-71> ip dhcp excluded-address 130.146.71.1 130.146.71.15
r22-71> ip dhcp pool CLIENT
r22-71/dhcp> import all
```

```
r22-71/dhcp> network 130.146.71.0 255.255.255.0
r22-71/dhcp> default-router 130.146.71.4
r22-71/dhcp> option 49 ip 130.146.59.5
r22-71/dhcp> lease 0 2
r22-71/dhcp> ^Z
r22-71>
```

**Note**

Details of the IOS interaction vary according to the specific release of IOS, the model of router and the hardware installed in the router.

DHCP option 49, the standard option of the X Window Display Manager, identifies [130.146.59.5](#) as the address of a Sun Ray server. In the absence of [AltAuth](#) and [Auth-Srvr](#) vendor-specific options, the client tries to find a Sun Ray server by broadcasting on the local subnet. If the broadcasts evoke no response, the client uses the address supplied in [t](#) option of the X Window Display Manager.

**Note**

This example is an unorthodox use of the option of the *X Window Display Manager*, but in a remote subnet deployment where vendor-specific options can not be delivered, it might be the only way of putting a client in touch with a server.

2. Configure the Sun Ray server to accept client connections from shared subnets by running `utadm -L` on.

```
# /opt/SUNWut/sbin/utadm -L on
### Turning on Sun Ray LAN connection
NOTE: utstart must be run before LAN connections will be allowed
#
```

3. Restart Sun Ray services on the Sun Ray server by issuing the `utstart` command to fully activate Sun Ray services on the shared subnet.

```
# /opt/SUNWut/sbin/utstart
A warm restart has been initiated... messages will be logged to /var/opt/SUNWut/log/messages.
```

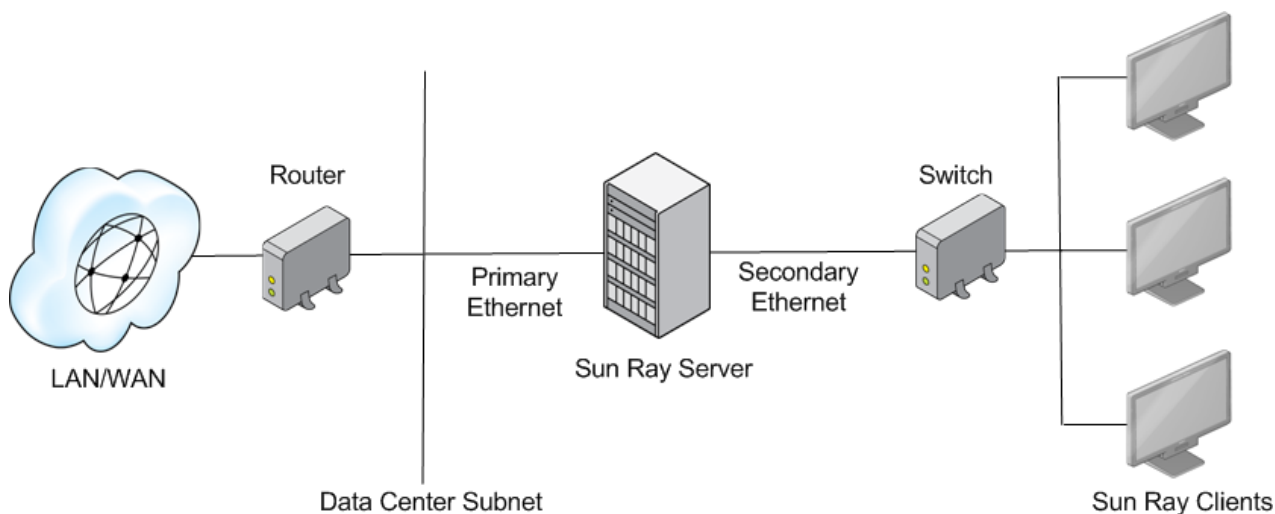
19.4. Using a Private Network Configuration

This section provides information about using a private network configuration for a Sun Ray environment. This is a supported configuration, but it adds more complexity to the initial and ongoing Sun Ray network administration.

A private network configuration meets the following criteria:

- A private network segment is connected to one of the interfaces of the Sun Ray server.
- The Sun Ray server handles all the Sun Ray Client's IP and device configurations.
- The Sun Ray server defines the subnet's characteristics (such as the IP range and subnet).
- Only Sun Ray Clients are connected to the subnet.
- The Sun Ray server provides DHCP services.
- The private network is configured by using the `utadm -a interface` command.

[Figure 19.2, "Private Network Example"](#) shows an example of a private network configuration.

Figure 19.2. Private Network Example

19.4.1. Private Network Configuration Worksheet

Fill out [Table 19.2, “Private Network Configuration Worksheet”](#), so that the information is readily available during the actual configuration process. This worksheet is for configuring a Sun Ray server in a private network.

- Values that are provided in *italics* are only examples and should *not* be used.
- Values provided in normal font are defaults and can be used.
- Superscripted numbers ^(#) refer to footnotes at the end of each section.



Note

The blank rows in the worksheets are provided for you to add additional information about your environment if you choose to print the worksheets.

Table 19.2. Private Network Configuration Worksheet

Aspect or Variable	Default Value, Example, or (Other)	Your Primary Server Value	Your Secondary Server Value
Configuring the Sun Ray interconnect interface (Provide the start time) using <code>utadm</code>			
Interface name	<i>hme1</i> (Oracle Solaris), <i>eth1</i> (Oracle Linux)		
• Host address ⁽¹⁾	192.168.128.1		
• Net mask	255.255.255.0		
• Net address	192.168.128.0		
• Host name ⁽¹⁾	<i>hostname-interface-name</i>		
If the Sun Ray server is used for IP address allocation:			
• First Sun Ray Client address	192.168.128.16		

Aspect or Variable	Default Value, Example, or (Other)	Your Primary Server Value	Your Secondary Server Value
• Number of Sun Ray Client addresses ⁽²⁾	X		
Firmware server ⁽³⁾	192.168.128.1		
Router ⁽³⁾	192.168.128.1		
Specify additional server list? (optional)	(yes or no)		
• If yes, filename	<i>filename</i>		
• Or, Server IP address	192.168.128.2		
Configuring Sun Ray Software using <i>utconfig</i> (Provide the start time)			
Admin password	<i>adminpass</i>		
Configure Admin GUI? If yes, then:			
• Apache Tomcat installation directory	<i>/opt/apache-tomcat</i>		
• Sun Ray admin server port number	1660		
• Enable remote administration? (optional)	(yes or no)		
• Enable secure connection? (optional)	(yes or no)		
Configure Kiosk Mode? (optional)	(yes or no)		
• If yes, User prefix	<i>utku</i>		
• Group name	<i>utkiosk</i>		
• User ID range start	150000		
• Number of users ⁽⁴⁾	25		
Configure failover group? (optional)	(yes or no)		
• If yes, Failover group signature ⁽⁵⁾	<i>signature1</i>		

⁽¹⁾ These values are different for each Sun Ray server, even if that server is part of a failover group.

⁽²⁾ These values must be unique among the servers in a failover group. The following guidelines can help you determine what addresses to allocate for each Sun Ray server:

- $X = (\text{Number of clients} / (\text{Number of servers} - 1)) - 1$
- First unit address for primary server = 192.168.128.16
- Last unit address for all servers = $X + \text{first unit address}$. If last unit address is greater than 240, reduce to 240.
 - First unit address for secondary servers = $1 + \text{last unit address of previous server}$. If first unit address is greater than 239, configure for a class B network. Example: 120 clients, 4 servers. $X = 39$

⁽³⁾ These values are the same as the interface host address by default.

⁽⁴⁾ The value entered for the number of users is the greater of:

- The total number of Sun Ray Clients
- The total number of disconnected and active sessions

⁽⁵⁾ This signature *must* be the same for every Sun Ray server in a failover group. The signature requires at least one numeric character.

19.4.2. How to Configure a Sun Ray Server in a Private Network

This procedure shows how to configure a Sun Ray server in a private network.

1. Log in as the superuser of the Sun Ray server, either locally or remotely.

**Note**

Make sure that the `/etc/hosts` file contains the IP address of the system host name.

2. Configure the Sun Ray interconnect interface:

```
# /opt/SUNWut/sbin/utadm -a interface-name
```

where *interface-name* is the name of the interface to the Sun Ray interconnect, for example: `hme1`, `qfe0`, or `ge0` (Oracle Solaris) or `eth1` (Oracle Linux).

The `utadm` script begins configuring DHCP for the Sun Ray interconnect, restarts the DHCP daemon, and configures the interface. The script then lists the default values and asks whether they are acceptable.

**Note**

If the IP addresses and DHCP configuration data are not set up correctly when the interfaces are configured, the failover feature cannot work properly. In particular, configuring the Sun Ray server's interconnect IP address as a duplicate of any other server's interconnect IP address may cause the Sun Ray Authentication Manager to generate Out of Memory errors.

3. Evaluate the default values:

- If you are satisfied with the default values, and the server is not part of a failover group, answer `y`.
- Otherwise, answer `n` and accept whatever default values are shown by pressing Return, or provide the correct values from the worksheet.

The `utadm` script prompts for the following:

- New host address (192.168.128.1)
- New netmask (255.255.255.0)
- New host name (*hostname-interface-name*)
- Offer IP addresses for this interface? ([Y]/N)
- New first Sun Ray Client address (92.168.128.16)
- Total number of Sun Ray Client address (*x*)
- New authorization server address (192.168.128.1)
- New firmware server address (192.168.128.1)
- New router address (192.168.128.1)
- An additional server list.

If you answer yes, it requests either a file name (*filename*) or a Server IP Address (192.168.128.2).

4. The `utadm` script again lists the configuration values and asks whether they are acceptable.

- If not, answer n and revise the answers provided in Step 3.
- If the values are correct, answer y. The following Sun Ray files are configured:

For Oracle Solaris:

```
/etc/hostname.interface-name
/etc/inet/hosts
/etc/inet/netmasks
/etc/inet/networks
```

For Oracle Linux:

```
/etc/opt/SUNWut/net/dhcp/SunRay-options
/etc/opt/SUNWut/net/dhcp/SunRay-interface-eth1
/etc/opt/SUNWut/net/hostname.eth1
/etc/hosts
/etc/opt/SUNWut/net/netmasks
/etc/opt/SUNWut/net/networks
/etc/dhcpd.conf
```

The `utadm` script configures the Sun Ray Client firmware versions and restarts the DHCP daemon.

5. Repeat this procedure for each of the secondary servers in your failover group.

19.4.3. How to List the Current Network Configuration

```
# utadm -l
```

19.4.4. How to Print a Private Network Configuration

```
# utadm -p
```

For each interface, this command displays the host name, network, netmask, and number of IP addresses assigned to Sun Ray Clients by DHCP.



Note

Sun Ray servers require static IP addresses; therefore, they cannot be DHCP clients.

19.4.5. How to Delete an Interface

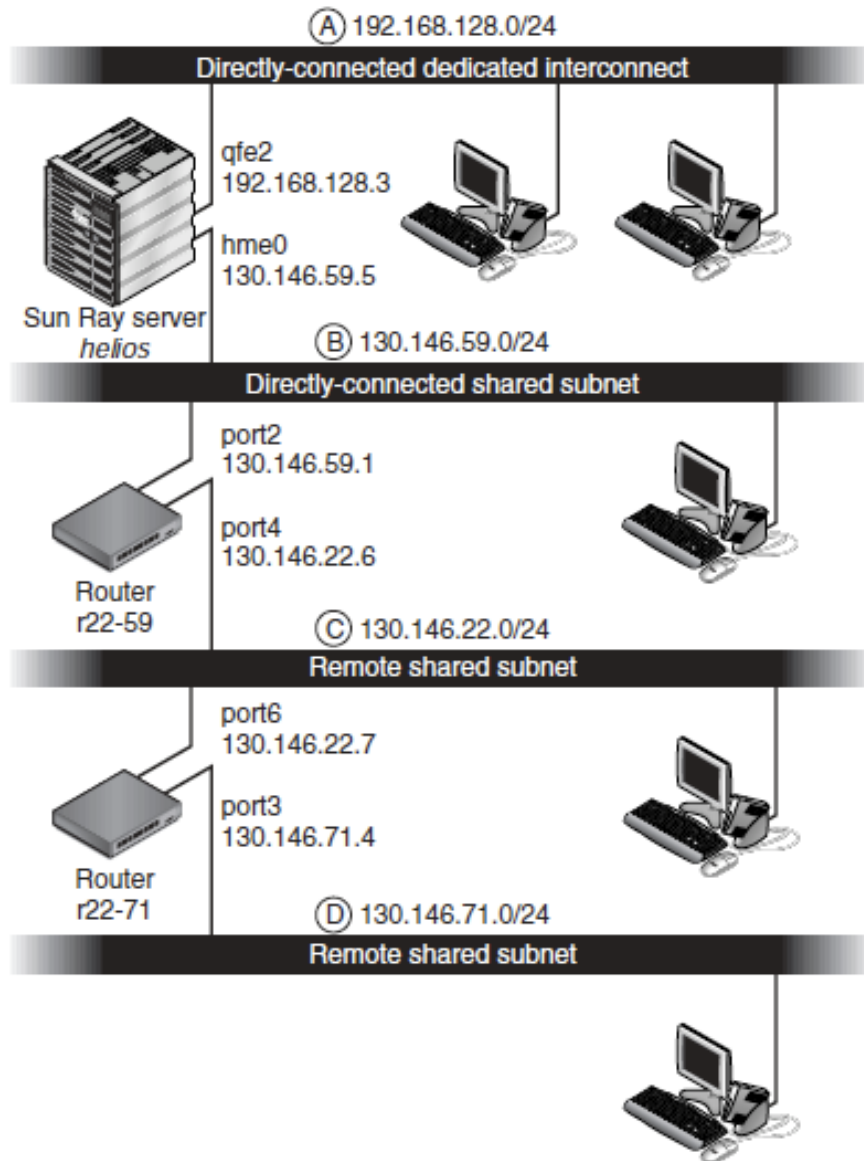
```
# utadm -d interface_name
```

This command deletes the entries that were made in the `hosts`, `networks`, and `netmasks` files and deactivates the interface as a Sun Ray interconnect.

19.4.6. Example Private Network Setup

The following section presents an example of a Sun Ray Client deployment on the private network interconnect A (directly-connected dedicated interconnect) as shown in [Figure 19.3, “Example of Alternate Private Network Topology”](#).

Figure 19.3. Example of Alternate Private Network Topology



Subnet A is a private network. Its subnet will use IP addresses in the range [192.168.128.0/24](#). The Sun Ray server named *helios* is attached to the interconnect through its *qfe2* network interface, which will be assigned the IP address [192.168.128.3](#).

In an interconnect scenario, the DHCP service on the Sun Ray server always provides both basic networking parameters and additional configuration parameters to the Sun Ray Client. The answers to the three predeployment questions are as follows:

- **From which DHCP server will clients on this subnet get their basic IP networking parameters?**

On a directly connected dedicated interconnect, basic networking parameters are always supplied by the DHCP service on the Sun Ray server.

- **From which DHCP server will clients on this subnet get additional configuration parameters to support features such as firmware download?**

On a directly connected dedicated interconnect, additional configuration parameters are always supplied by the DHCP service on the Sun Ray server.

- **How will clients on this subnet locate their Sun Ray server?**

On a directly connected dedicated interconnect, the Sun Ray Client is always notified of the location of the Sun Ray server through an additional configuration parameter supplied when Sun Ray services are restarted.

This example shows the DHCP service for the directly connected dedicated interconnect A shown in [Figure 19.3, "Example of Alternate Private Network Topology"](#).

1. Configure the Sun Ray server to provide both basic and additional parameters to the interconnect.

Use the `utadm -a interface-name` command to configure DHCP service for clients on an interconnect. In this example, the interconnect is attached through interface `qfe2`:

```
# /opt/SUNWut/sbin/utadm -a qfe2
### Configuring /etc/nsswitch.conf
### Configuring Service information for Sun Ray
### Disabling Routing
### configuring qfe2 interface at subnet 192.168.128.0
Selected values for interface "qfe2"
host address: 192.168.128.1
net mask: 255.255.255.0
net address: 192.168.128.0
host name: helios-qfe2
net name: SunRay-qfe2
first unit address: 192.168.128.16
last unit address: 192.168.128.240
auth server list: 192.168.128.1
firmware server: 192.168.128.1
router: 192.168.128.1
Accept as is? ([Y]/N): n
new host address: [192.168.128.1] 192.168.128.3
new netmask: [255.255.255.0]
new host name: [helios-qfe2]
Do you want to offer IP addresses for this interface? ([Y]/N):
new first Sun Ray address: [192.168.128.16]
number of Sun Ray addresses to allocate: [239]
new auth server list: [192.168.128.3]
To read auth server list from file, enter file name:
Auth server IP address (enter <CR> to end list):
If no server in the auth server list responds, should an auth server be located by
broadcasting on the network? ([Y]/N):
new firmware server: [192.168.128.3]
new router: [192.168.128.3]
Selected values for interface "qfe2"
host address: 192.168.128.3
net mask: 255.255.255.0
net address: 192.168.128.0
host name: helios-qfe2
net name: SunRay-qfe2
first unit address: 192.168.128.16
last unit address: 192.168.128.254
auth server list: 192.168.128.3
firmware server: 1 192.168.128.3
router: 192.168.128.3
Accept as is? ([Y]/N):
### successfully set up "/etc/hostname.qfe2" file
### successfully set up "/etc/inet/hosts" file
```

```

### successfully set up "/etc/inet/netmasks" file
### successfully set up "/etc/inet/networks" file
### finished install of "qfe2" interface
### Building network tables - this will take a few minutes
### Configuring firmware version for Sun Ray
All the units served by "helios" on the 192.168.128.0
network interface, running firmware other than version
"2.0_37.b,REV=2002.12.19.07.46" will be upgraded at their
next power-on.
### Configuring Sun Ray Logging Functions
DHCP is not currently running, should I start it? ([Y]/N):
### started DHCP daemon
#

```

In this example, the default values initially suggested by `utadm` were not appropriate. Specifically, the suggested value for the server's IP address on the interconnect was not the desired value. The administrator replied `n` to the first "Accept as is?" prompt and was given the opportunity to provide alternative values for the various parameters.

- Restart Sun Ray services on the Sun Ray server by issuing the `utstart` command to fully activate Sun Ray services on the newly defined interconnect.

```

# /opt/SUNWut/sbin/utstart
A warm restart has been initiated... messages will be logged to /var/opt/SUNWut/log/messages.

```

19.5. Sun Ray Client Initialization Requirements Using DHCP

Because Sun Ray Clients are stateless, they rely entirely on network services to provide the configuration data they need to complete their initialization.

- Each Sun Ray Client must first acquire basic network parameters, such as a valid IP address, on the network to which it is connected.
- The Sun Ray Client can also be supplied with additional configuration information to support advanced product features, such as the ability to update the Sun Ray Client firmware and to report exception conditions to a `syslog` service.
- The Sun Ray Client must locate and contact a Sun Ray server that can offer desktop services to the Sun Ray user.

The Sun Ray Client uses the Dynamic Host Configuration Protocol (DHCP) to obtain this information.

19.5.1. DHCP Basics

The Sun Ray Client is a DHCP client that solicits configuration information by broadcasting DHCP packets on the network. The requested information is supplied by one or more DHCP servers in response to the client's solicitations. DHCP service may be provided by a DHCP server process executing on a Sun Ray server, by DHCP server processes executing on other systems, or by some combination of the two. Any conforming implementation of a DHCP service can be used to satisfy the DHCP requirements of the Sun Ray Client. The Oracle Solaris DHCP service is one such implementation. Third-party implementations executing on non-Sun platforms can also be configured to deliver information to Sun Ray Clients.

The DHCP protocol defines a number of standard options that can be used to inform the client of a variety of common network capabilities. DHCP also allows a number of vendor-specific options that carry information that is meaningful only to individual products. For more information, see [Table 19.4, "Alternate Vendor-Specific DHCP Options"](#).

The Sun Ray Client depends on a small number of standard options to establish its basic network parameters. It depends on several standard and vendor-specific options to provide the additional

information that constitutes a complete client configuration. If these additional configuration parameters are not supplied, the client cannot perform certain activities, the most important of which is the downloading of new Sun Ray Client firmware. [Table 19.4, “Alternate Vendor-Specific DHCP Options”](#) lists the vendor-specific options.



Note

If an administrator chooses not to make this additional configuration information available to the Sun Ray Clients, a procedure must be established to deliver firmware updates to them. One solution would be a small, dedicated interconnect on one Sun Ray server. Then, the administrator can transfer the clients one-by-one when new firmware becomes available on the server, for example, through a patch or Sun Ray product upgrade.

The location of the Sun Ray server is usually conveyed to the Sun Ray Client through one of a pair of DHCP vendor-specific options, [AuthSrvr](#) and [AltAuth](#).

If the Sun Ray Client does not receive this information, it uses a broadcast-based discovery mechanism to find a Sun Ray server on its subnet. If the broadcast-based discovery mechanism fails, the Sun Ray Client interprets the DHCP standard option (option 49) of the X Window Display Manager as a list of Sun Ray server addresses where it attempts to contact Sun Ray services. This feature can simplify the DHCP configuration of LAN-deployed Sun Rays by removing the need for a DHCP vendor option to carry this information.

[Table 19.3, “DHCP Service Parameters Available”](#) provides the list of available DHCP service parameters.

Table 19.3. DHCP Service Parameters Available

Parameters	Sun Ray Server DHCP Service	External DHCP Service With Vendor-Specific Options	External DHCP Service Without Vendor-Specific Options	No DHCP Service
Basic network parameters	Yes	Yes	Yes	No
Additional parameters (for firmware download, etc.)	Yes	Yes	No	No
Sun Ray server location	Yes	Yes	Yes, through broadcast discovery or the X Display Manager standard option	Yes, through broadcast discovery

19.5.2. DHCP Parameter Discovery

DHCP enables two stages of parameter discovery. The initial [DHCPDISCOVER](#) stage discovers basic network parameters. This stage may be followed by a [DHCPINFORM](#), which finds additional information that was not provided during [DHCPDISCOVER](#).

All Sun Ray Clients must have access to at least one DHCP service, which provides network parameters in response to a [DHCPDISCOVER](#) request from the client. Sun Ray Clients can exploit the [DHCPINFORM](#) feature, which enables full configuration of the client, even when an external DHCP service that is not capable of providing complete configuration data provides the network parameters of the Sun Ray Client.

19.5.3. DHCP Relay Agent

The Sun Ray Client sends DHCP requests as broadcast packets that propagate only on the local LAN segment or subnet. If the Sun Ray Client resides on the same subnet as the DHCP server, the DHCP server can see the broadcast packet and respond with the information the Sun Ray Client needs. If the Sun Ray Client resides on a different subnet than the DHCP server, the client must depend on a local DHCP Relay Agent to collect the broadcast packet and forward it to the DHCP server. Depending on the physical network topology and DHCP server strategy, the administrator might need to configure a DHCP Relay Agent on each subnetwork to which Sun Ray clients are connected. Many IP routers provide DHCP Relay Agent capability. If a deployment plan requires the use of a DHCP Relay Agent and the administrator decides to activate this capability on a router, the appropriate instructions can be found in the router documentation, usually under the heading of "DHCP Relay" or "BOOTP forwarding." DHCP is derived from an earlier protocol called BOOTP. Some documentation uses these names interchangeably.

In certain cases, an existing enterprise DHCP service provides the Sun Ray Client with its IP address while a Sun Ray server provides it with firmware version details and Sun Ray server location. If a deployment plan calls for DHCP parameters to be provided to the client by multiple servers, and none of those servers is connected to the subnet where the client resides, the DHCP Relay Agent should be configured so that the clients subnet can deliver broadcasts to all the DHCP servers. For example, in routers controlled by a Cisco IOS Executive, the `ip helper-address` command activates a DHCP Relay Agent. Specifying multiple arguments to the `ip helper-address` command enables relaying to multiple DHCP servers. For more information, see [Section 19.3.5.2, "Deployment on a Remote Subnet"](#).

19.5.4. Simplifying DHCP Configuration of Remote Sun Ray Clients

You can simplify the DHCP configuration of Sun Ray Clients at remote sites by using the *X Window System Display Manager* option to supply a list of available Sun Ray servers. This option eliminates the need for Sun Ray vendor options as well as the need to forward DHCPINFORM requests to a Sun Ray server.

For a more complete treatment of network configuration, including DHCP and vendor-specific options, see [Table 19.3, "DHCP Service Parameters Available"](#) and [Table 19.4, "Alternate Vendor-Specific DHCP Options"](#).

The following example is a sample DHCP configuration for a Cisco IOS-based router.

```
ip dhcp excluded-address 129.149.244.161
ip dhcp pool CLIENT
import all network 129.149.244.160 255.255.255.248
default-router 129.149.244.161
option 26 hex 0556
option 49 ip 10.6.129.67 129.146.58.136
lease 0 2
```

Option 49, the *X Window System Display Manager* option, lists IP addresses [10.6.129.67](#) and [129.146.58.136](#) as Sun Ray servers. The Sun Ray Client tries to connect to those servers when it receives a DHCP response from the router. Option 26 sets the Maximum Transmission Unit (MTU), which defines the maximum packet size for the Sun Ray connections, in this case, 1366 bytes rather than the default Ethernet MTU of 1500 bytes. This setting is necessary to provide space for the IPSec headers to implement a virtual private network (VPN) connection.

The DHCP service, either directly from an ISP or from a home firewall, is also required to assign the router its IP address behind the firewall.

The router's WAN port either plugs directly into the DSL/Cable modem or into the home firewall or gateway. The Sun Ray Client then plugs into one of the four LAN ports on the router. A VPN router

plugged directly into the DSL or cable modem can be connected only to a Sun Ray Client. If the router has been configured to supply DHCP parameters to the Sun Ray Client, it will instruct the client to try to connect to the appropriate Sun Ray server.

The router should start a VPN tunnel when it is plugged in, which it should always be on. Each router should be connected to the VPN gateway and programmed with a user name based on an user's ID and a random password. The VPN gateway should be configured to allow only Sun Ray traffic to pass, and only to a limited number of hosts, so that users cannot connect anything else to the LAN side of the router and then connect into the corporate network. However, users may connect more than one Sun Ray Client.

Whenever a VPN or other tunnel is being used, you need to take account of the IP MTU across the path between the server and the Sun Ray Client. The VPN typically packs additional control data into each packet, which reduces the available space for application data.

The latest Sun Ray firmware attempts to compensate for this reduction automatically, but this process is not always possible. Make sure that the Sun Ray Client has the latest firmware. Installing the latest patch on the server is not sufficient. You must also make sure that the client was configured to update its firmware and then check that the update occurred.

If the Sun Ray Client has the latest firmware but the problem still occurs, then the client must be set to work with a reduced MTU. You can update the client through whatever mechanism you use to give the Sun Ray its basic configuration data, such as DHCP, TFTP or, if the client is running GUI-capable firmware, local configuration on the Sun Ray Client itself.

The site should know what the effective MTU is across the VPN. If not, see any available technical archives or the ThinkThin blog on <http://blogs.oracle.com/ThinkThin/>. If a precise MTU is not important, then a low estimate, such as 1350 (the standard value is 1500), should be sufficient to let you verify that MTU is the cause of the problem.

After you update and restart the Sun Ray Client, the client reports the new MTU value to the server, and the server adjusts its packet-construction strategy to fit within that MTU. The client should no longer send Sun Ray traffic that is too big to be delivered in one piece through the VPN tunnel.

**Note**

Local settings on the Sun Ray Client generally override values obtained from other sources, such as `.parms` files or DHCP. Therefore, you must provide the ability to clear a setting so that the value from a `.parms` file is not overridden and can be used for configuration. For numeric values, include an empty field. For switch settings, click the Clear button when modifying a setting. The `utquery` output from a client reflects the values that are defined in the local configuration.

19.5.5. Standard DHCP Parameters

A set of Sun Ray Clients can be started with only standard DHCP parameters, shifting the burden of defining the server list to the Domain Name Service (DNS) and firmware management to TFTP.

If `sunray-config-servers` and `sunray-servers` are defined appropriately by the DNS serving a set of remote Sun Rays Clients, no extra DHCP parameters are required other than basic network information.

- A DNS client incorporated in the firmware allows many values to be names rather than IP addresses. Most values can be either a name or an IP address. If a name is specified, the DNS lookup appends the configured domain name. Components are stripped successively until the lookup succeeds or only two components are left in the domain name. If none of those lookups succeed, the name is looked up by

itself. If the name itself ends with a dot character ("."), the name is taken to be a rooted name, and it is looked up without domain name components appended.

- DHCP option 66 (TFTP server name) is supported as an alternative to the `FWSrvr` vendor-specific option listed in [Section 19.5.6, “Vendor-specific DHCP Options”](#). This string value must either be a single IP address or a DNS host name that resolves to a list of IP addresses. If it is a list of IP addresses, one is chosen randomly.
- A firmware maintenance mechanism creates `*.parms` files in the TFTP home directory (one for each model type), which are read in lieu of using the `NewTVer` DHCP vendor option. Thus, remote firmware upgrades are possible without DHCP access to the `NewTVer` value. The `*.parms` files contain the version, hardware revision, and barrier levels, eliminating unnecessary file reads in cases where the barrier would have prevented writing the firmware to flash memory. For details on options that can be used to configure the `.parms` files, see the `utfwadm` man page.
- A default DNS name for the firmware server, `sunray-config-servers`, is used when neither option 66 nor `FWSrvr` is given. Defining this name in DNS provides the firmware server address without DHCP options, just DNS servers and domain name.
- Inclusion of `servers=_server name list_` and `select=inorder|random` in the `*.parms` files enables specification of a list of server names and specification of whether the names should be used in order, or at random. If a name resolves to multiple addresses, an IP address is chosen according to the select keyword.
- When neither a server list nor an `AltAuth` list is provided, the default name

`sunray-servers` is looked up in DNS and the list of IP addresses is used in place of the `AltAuth` list.

In the event of an error in the firmware download, error messages provide additional information that can be useful in diagnosing and correcting the problem. See [Chapter 16, *Troubleshooting Icons*](#).

Also, during DNS lookups, a status line in the OSD icon shows the name being looked up and, if one is found, the IP address.

19.5.6. Vendor-specific DHCP Options

[Table 19.4, “Alternate Vendor-Specific DHCP Options”](#) lists the vendor-specific DHCP options that Sun Ray defines and uses.

Table 19.4. Alternate Vendor-Specific DHCP Options

Option Code	Parameter Name	Client Class	Data Type	Required?	Granularity	Max Count	Comments
21	<code>AuthSrvr</code>	<code>SUNW.NewT.SUNW</code>	IP	Mandatory	1	1	Single Sun Ray server IP addresses
22	<code>AuthPort</code>	<code>SUNW.NewT.SUNW</code>	NUMBER	Optional	2	1	Sun Ray server port
23	<code>NewTVer</code>	<code>SUNW.NewT.SUNW</code>	ASCII	Optional	1	0	Desired firmware version
24	<code>LogHost</code>	<code>SUNW.NewT.SUNW</code>	IP	Optional	1	1	Syslog server IP address

Vendor-specific DHCP Options

Option Code	Parameter Name	Client Class	Data Type	Required?	Granularity	Max Count	Comments
25	LogKern	SUNW.NewT.SUNW	NUMBER	Optional	1	1	Log level for kernel
26	LogNet	SUNW.NewT.SUNW	NUMBER	Optional	1	1	Log level for network
27	LogUSB	SUNW.NewT.SUNW	NUMBER	Optional	1	1	Log level for USB
28	LogVid	SUNW.NewT.SUNW	NUMBER	Optional	1	1	Log level for video
29	LogAppl	SUNW.NewT.SUNW	NUMBER	Optional	1	1	Log level for firmware application
30	NewTBW	SUNW.NewT.SUNW	NUMBER	Optional	4	1	Bandwidth cap, value is bits per second
31	FWSrvr	SUNW.NewT.SUNW	IP	Optional	1	1	Firmware TFTP server IP address
32	NewTDispIndx	SUNW.NewT.SUNW	NUMBER	Optional	4	1	Obsolete. Do not use.
33	Intf	SUNW.NewT.SUNW	ASCII	Optional	1	0	Sun Ray server interface name
34	NewTFlags	SUNW.NewT.SUNW	NUMBER	Optional	4	1	Obsolete. Do not use.
35	AltAuth	SUNW.NewT.SUNW	IP	Optional	1	0	List of Sun Ray server IP addresses
36	BarrierLevel	SUNW.NewT.SUNW	NUMBER	Mandatory	4	1	Firmware Download: barrier level

The client can perform its basic functions even if none of these options are delivered during initialization, but some advanced client features do not become active unless certain options are delivered to the client. In particular:

- [AltAuth](#) and [AuthSrvr](#) indicate the IP addresses of Sun Ray servers. Addresses in the [AltAuth](#) list are tried in order until a connection is established. Current firmware ignores [AuthSrvr](#) if [AltAuth](#) is provided, but always specify [AuthSrvr](#) for the benefit of old (pre Sun Ray Server Software 1.3) firmware, which cannot handle the [AltAuth](#) option. If neither of these options is supplied, the client tries to locate a Sun Ray server by sending broadcasts on the local subnet. The client tries to contact a Sun Ray server at the address supplied in the option of the X Window Display Manager if that option has been provided.
- [NewTVer](#) and [FWSrvr](#) must both be provided in order for the client to attempt a firmware download. [NewTVer](#) contains the name of the firmware version that the client should use. If this name does not

match the name of the firmware version that the client is actually running, the client tries to download the desired firmware from a TFTP server at the address given by `FWSrvr`.

- `LogHost` must be specified in order for the client to report messages through the syslog protocol. Reporting thresholds for major client subsystems are controlled by the `LogKern`, `LogNet`, `LogUSB`, `LogVid`, and `LogAppl` options.



Note

Because the message formats, contents, and thresholds are intended for use only by service personnel, they are not documented here.

The DHCP Client Class name for all Sun Ray vendor-specific options is `SUNW.NewT.SUNW`. The client cites this name in DHCP requests so that the server can respond with the appropriate set of vendor-specific options. This mechanism guarantees that the client is not sent vendor options defined for some other type of equipment and that other equipment is not sent options that are meaningful only to the client.

19.5.7. Encapsulated Options

For each parameter name, there is a vendor ID, an option code, an option type, and an indication as to whether the parameter is mandatory.

Vendor-specific options are delivered through encapsulated options in DHCP. Encapsulated options are somewhat more complicated, as illustrated in the following DHCPINFORM response, or DHCPACK, which shows the taxonomy of the bytes in the vendor-specific information portion.

```
2b 4a 17 1d 32 2e 30 .....: .+J..2.0
0140 5f 31 39 2e 63 2c 52 45 56 3d 32 30 32 2e 30 _19.c,RE V=2002.0
0150 39 2e 30 36 2e 31 35 2e 35 34 21 04 68 6d 65 30 9.06.15. 54!.hme0
0160 1f 04 81 92 3a 88 15 04 81 92 3a 88 1d 01 06 1c ....:... ..:....
0170 01 06 1b 01 06 1a 01 06 19 01 06 18 04 81 92 3a .....: .....:
0180 88 16 02 1b 61
```



Note

In this description, hexadecimal values are preceded by 0x and followed by their decimal value, after an = sign, as in `0x2b=43`.

- The first byte is the option code.
- The next byte represents the encapsulated option length, that is, the number of bytes that make up the option value.
- The next one or more bytes make up the multi-byte option value.

The option value is followed by another encapsulated option code, and so on.

The example begins with `0x2b=43`, the DHCP option for vendor-specific information. It has a length of `0x4a=74` bytes, which is the total number of bytes that follow. These bytes contain the encapsulated vendor options.

The remainder of the example represents the value of the vendor-specific information options. The first byte contains the first encapsulated option, whose value is `0x17=23`, and the `NewTVer` option, whose value type is ASCII. The next byte is `0x1d=29`, which is the length of the `NewTVer` string. These options are followed by 29 bytes that represent the string itself.

The ASCII interpretation at the right of the DHCPACK, is `2.0_19.c,REV=2002.09.06.15.54`. This is the end of the first encapsulated option. The next byte is the beginning of the next option, `Intf`, represented

by `0x21=33`. The next byte, the length, is `0x04=4`, and the next four bytes are the ASCII value `hme0`. That's the end of the second encapsulated option.

The next byte is `0x1f=31`, which represents the FWSrvr parameter, whose function is to indicate the IP address of the firmware TFTP server. The next byte is the length, `4`, which is always be true for an IP address. The hexadecimal value is `0x81 0x92 0x3a 0x88`, which corresponds to the IP address `129.146.58.136`.

19.6. Failover Groups

This section provides information about configuring a failover group in a network configuration when using a private network or when using a Sun Ray server as a DHCP server.

If you have Sun Ray dedicated interconnects, all services required by Sun Ray clients should be provided by multiple, redundant servers to ensure continuity of Sun Ray service in the case of a network or system failure. For example, you must configure DHCP (IP address assignment and configuration) or DNS (name resolution) on all servers.



Note

The failover feature cannot work properly if the IP addresses and DHCP configuration data are not set up properly when the interfaces are configured. In particular, if any Sun Ray server's interconnect IP address is a duplicate of any other server's interconnect IP address, the Sun Ray Authentication Manager will fail to operate properly.

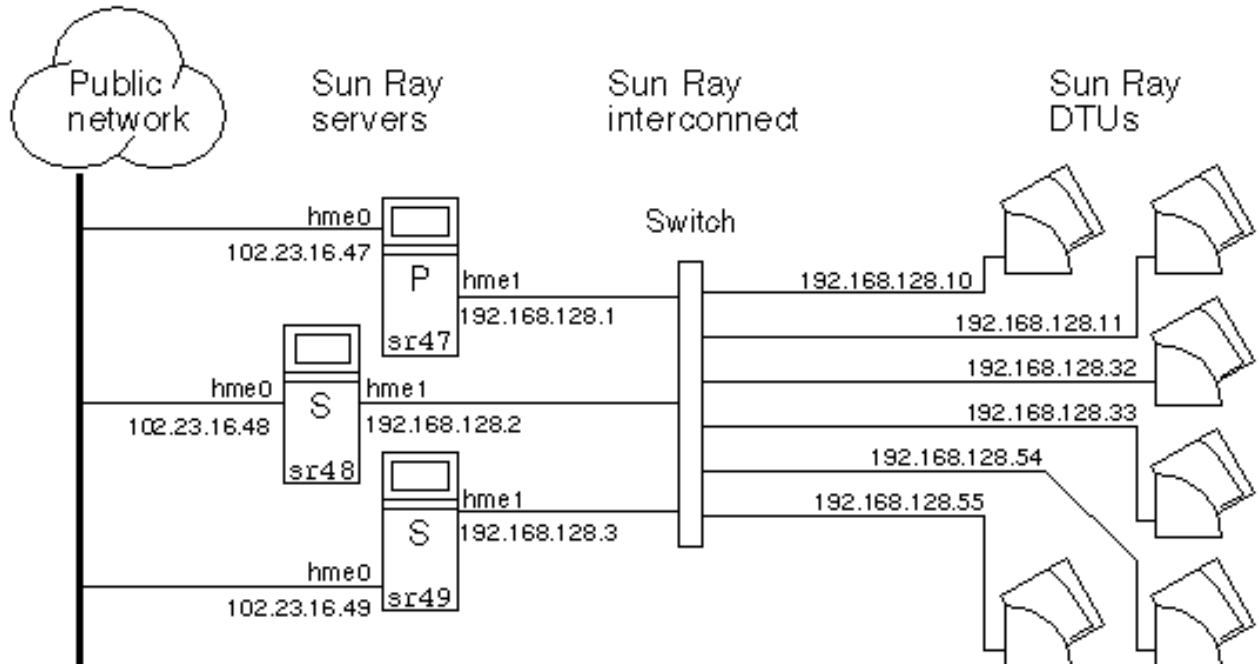
19.6.1. Network Topologies

A failover group can consist of servers in either a common, dedicated interconnect or servers within a LAN. However, the servers in a failover group must still be able to reach one another, using multicast or broadcast, over at least one shared subnet. Servers in a group authenticate (or "trust") one another using a common group signature. The group signature is a key used to sign messages sent between servers in the group. This key must be configured to be identical on each server.

When a dedicated interconnect is used, all servers in the failover group should have access to, and be accessible by, all the Sun Ray Clients on a given sub-net. Routers should not be attached to a dedicated interconnect. The failover environment supports the same interconnect topologies that are supported by a single-server Sun Ray environment; however, switches should be multicast-enabled.

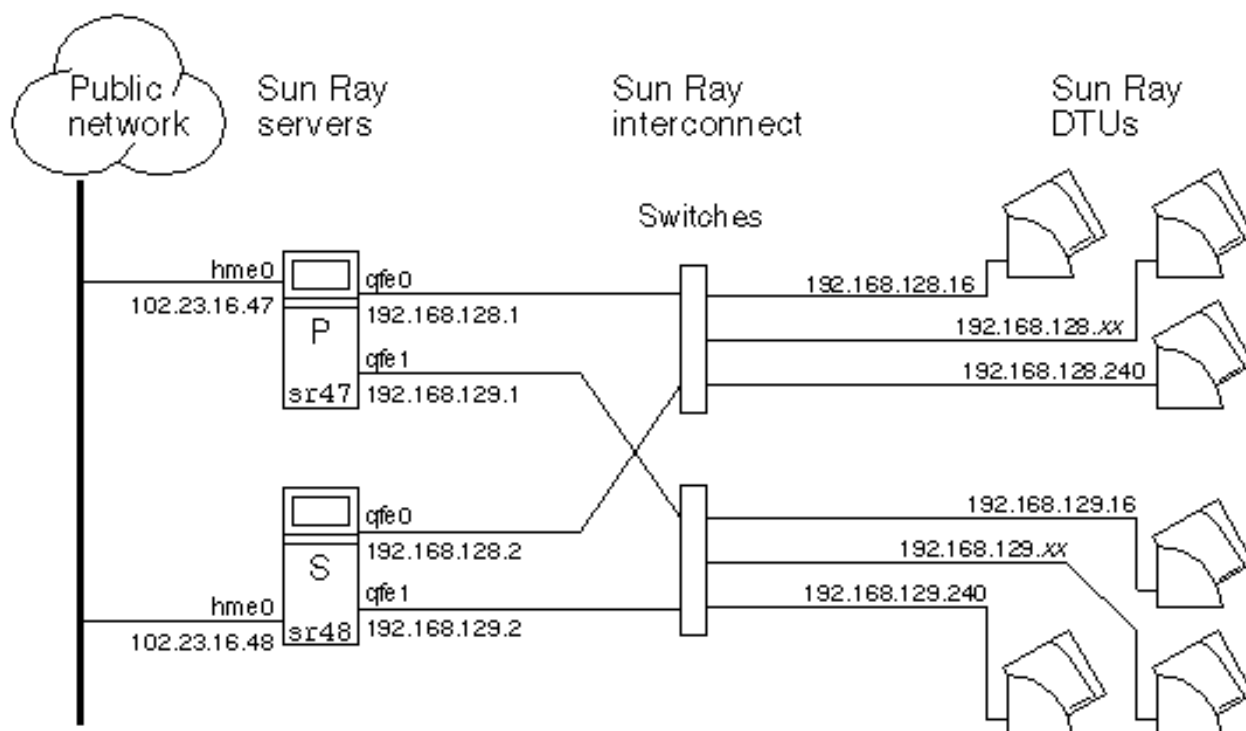
If multicast does not work in your network, you may use broadcast instead. To disable multicast, use the `enableMulticast` property in the `auth.props` file. In special cases, you may configure an explicit list of group servers using `utgmtarget`. For example, you would use `utgmtarget` to integrate servers on a different subnet into a failover group. Communication to these servers use unicast. Note that adding such a server to the group will require a restart of the entire server group.

Figure 19.4, "Simple Failover Group" shows a simple failover group setup.

Figure 19.4. Simple Failover Group

When a server in a failover group fails for any reason, each Sun Ray Client connected to that server reconnects to another server in the same failover group. The failover occurs at the user authentication level: the client connects to a previously existing session for the user's token. If session exists, the client connects to a server selected by the load-balancing algorithm. This server then presents a login screen to the user, and the user must relogin to create a new session. The state of the session on the failed server is lost.

Figure 19.5, "Redundant Failover Group" shows an example of a redundant failover group.

Figure 19.5. Redundant Failover Group

The redundant failover group, shown in the illustration above, can provide maximum resources to a few Sun Ray Clients. The server sr47 is the primary Sun Ray server, and sr48 is the secondary Sun Ray server; other secondary servers (sr49, sr50, and so on) are not shown.

19.6.2. Setting Up IP Addressing

You can use the `utadm` command to set up a DHCP server. The default DHCP setup configures each interface for 225 hosts and uses private network addresses for the Sun Ray interconnect. For more information, see the `utadm` man page.

Before setting up IP addressing, you must decide upon an addressing scheme. The following examples discuss setting up class C and class B addresses.

19.6.2.1. Setting Up Server and Client Addresses

The loss of a server usually implies the loss of its DHCP service and its allocation of IP addresses. Therefore, more DHCP addresses must be available from the address pool than the number of Sun Ray Clients. Consider the situation of 5 servers and 100 Sun Ray Clients. If one of the servers fails, the remaining DHCP servers must have enough available addresses so that every "orphaned" Sun Ray Client is assigned a new working address.

Table 19.5, "Configuring 5 Servers for 100 clients" lists configuration settings used to configure 5 servers for 100 Sun Ray Clients, accommodating the failure of two servers (class C) or four servers (class B).

Table 19.5. Configuring 5 Servers for 100 clients

CLASS C (2 Servers Fail)			CLASS B (4 Servers Fail)	
Servers	Interface Address	Client Address Range	Interface Address	Client Address Range
serverA	192.168.128.1	192.168.128.16 to 192.168.128.49	192.168.128.1	192.168.128.16 to 192.168.128.116
serverB	192.168.128.2	192.168.128.50 to 192.168.128.83	192.168.129.1	192.168.129.16 to 192.168.129.116
serverC	192.168.128.3	192.168.128.84 to 192.168.128.117	192.168.130.1	192.168.130.16 to 192.168.130.116
serverD	192.168.128.4	192.168.128.118 to 192.168.128.151	192.168.131.1	192.168.131.16 to 192.168.131.116
serverE	192.168.128.5	192.168.128.152 to 192.168.128.185	192.168.132.1	192.168.132.16 to 192.168.132.116

The formula for address allocation is: address range (AR) = number of clients/(total servers - failed servers). For example, in the case of the loss of two servers, each DHCP server must be given a range of $100/(5-2) = 34$ addresses.

Ideally, each server would have an address for each client. This setup requires a class B network. Consider these conditions:

- If AR multiplied by the total number of servers is *less than or equal to* 225, configure for a class C network
- If AR multiplied by the total number of servers is *greater than* 225, configure for a class B network



Note

If all available DHCP addresses are allocated, a Sun Ray Client could request an address and still not find one available, perhaps because another unit has been allocated IP addresses by multiple servers. To prevent this condition, provide each DHCP server with enough addresses to serve all the Sun Ray Clients in a failover group.

19.6.2.2. Server Addresses

Server IP addresses assigned for the Sun Ray interconnect should all be unique. Use the `utadm` tool to assign them.

When the Sun Ray Client boots, it sends a DHCP broadcast request to all possible servers on the network interface. One or more servers respond with an IP address allocated from its range of addresses. The client accepts the first IP address that it receives and configures itself to send and receive at that address.

The accepted DHCP response also contains information about the IP address and port numbers of the Authentication Managers on the server that sent the response.

The client then tries to establish a TCP connection to an Authentication Manager on that server. If it is unable to connect, it uses a protocol similar to DHCP, in which it uses a broadcast message to ask the Authentication Managers to identify themselves. The client then tries to connect to the Authentication Managers that respond in the order in which the responses are received.

**Note**

For the broadcast feature to be enabled, the broadcast address (255.255.255.255) must be the last one in the list. Any addresses after the broadcast address are ignored. If the local server is not on the list, Sun Ray Clients cannot attempt to contact it.

Once a TCP connection to an Authentication Manager has been established, the client presents its token. The token is either a pseudo-token representing the individual client (its unique Ethernet address) or a smart card. The Session Manager then starts an X Window/X server session and binds the token to that session.

The Authentication Manager then sends a query to all the other Authentication Managers on the same subnet and asks for information about existing sessions for the token. The other Authentication Managers respond, indicating whether a session for the token exists and the last time the token was connected to the session.

The requesting Authentication Manager selects the server with the latest connection time and redirects the client to that server. If no session is found for the token, the requesting Authentication Manager selects the server with the lightest load and redirects the token to that server. A new session is created for the token.

The Authentication Manager enables both implicit (smart card) and explicit switching.

19.6.2.3. Configuring DHCP

In a large IP network, a DHCP server distributes the IP addresses and other configuration information for interfaces on that network.

19.6.2.3.1. Coexistence of the Sun Ray Server With Other DHCP Servers

The Sun Ray DHCP server can coexist with DHCP servers on other subnets, provided that you isolate the Sun Ray DHCP server from other DHCP traffic. Verify that all routers on the network are configured not to relay DHCP requests, which is the default behavior for most routers.

**Caution**

If the IP addresses and DHCP configuration data are not set up correctly when the interfaces are configured, the failover feature cannot work properly. In particular, configuring the Sun Ray server's interconnect IP address as a duplicate of any other server's interconnect IP address may cause the Sun Ray Authentication Manager to issue "Out of Memory" errors.

19.6.2.4. Administering Other Clients

If the Sun Ray server has multiple interfaces, one of which is the Sun Ray interconnect, the Sun Ray DHCP server should be able to manage both the Sun Ray interconnect and the other interfaces without cross-interference.

19.6.2.5. How to Set Up IP Addressing on Multiple Servers, Each with One Sun Ray Interface

1. Log in to the Sun Ray server as superuser and, open a shell window. Type:

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

where *interface_name* is the name of the Sun Ray network interface to be configured; for example, *hme[0-9]*, *qfe[0-9]*, or *ge[0-9]*. You must be logged on as superuser to run this command. The *utadm* script configures the interface (for example, *hme1*) at the subnet (in this example, 128).

The script displays default values, such as the following:

```
Selected values for interface "hme1"
host address: 192.168.128.1
net mask: 255.255.255.0
net address: 192.168.128.0
host name: serverB-hme1
net name: SunRay-hme1
first unit address: 192.168.128.16
last unit address: 192.168.128.240
auth server list: 192.168.128.1
firmware server: 192.168.128.1
router: 192.168.128.1 |
```

The default values are the same for each server in a failover group. Certain values must be changed to be unique to each server.

2. When you are asked to accept the default values, type **n**

```
Accept as is? ([Y]/N): n
```

3. Change the second server's IP address to a unique value, in this case 192.168.128.2:

```
new host address: [192.168.128.1] 192.168.128.2 |
```

4. Accept the default values for netmask, host name, and net name:

```
new netmask: [255.255.255.0]
new host name: [serverB-hme1]
```

5. Change the client address ranges for the interconnect to unique values. For example:

```
Do you want to offer IP addresses for this interface? [Y/N]:
new first Sun Ray address: [192.168.128.16] 192.168.128.50
number of Sun Ray addresses to allocate: [205] 34
```

6. Accept the default firmware server and router values:

```
new firmware server: [192.168.128.2]
new router: [192.168.128.2]
```

The **utadm** script asks if you want to specify an authentication server list:

```
auth server list: 192.168.128.1
To read auth server list from file, enter file name:
Auth server IP address (enter <CR> to end list):
If no server in the auth server list responds, should an auth server be located by
broadcasting on the network? ([Y]/N):
```

These servers are specified by a file containing a space-delimited list of server IP addresses or by manually entering the server IP addresses.

The newly selected values for interface **hme1** are displayed:

```
Selected values for interface "hme1"
host address: 192.168.128.2
net mask: 255.255.255.0
net address: 192.168.128.0
host name: serverB-hme1
net name: SunRay-hme1
first unit address: 192.168.128.50
last unit address: 192.168.128.83
auth server list: 192.168.128.1
```

```
firmware server: 192.168.128.2
router: 192.168.128.2
```

7. If these are correct, accept the new values:

```
Accept as is? ([Y]/N): y
```

8. Stop and restart the server and power cycle the clients to download the firmware.

For additional information, see the [utadm](#) man page.

19.6.3. Sun Ray Server Failover Group Worksheet

Fill out [Table 19.6, “Sun Ray Server Failover Group Worksheet”](#) and [Table 19.7, “First and Last Unit Address in a Failover Group”](#), so that the information is readily available during the actual configuration process.

- Values that are provided in *italics* are only examples and should *not* be used.
- Values provided in normal font are defaults and can be used.
- Superscripted numbers ^(#) refer to footnotes at the end of each section.



Note

The blank rows in the worksheets are provided for you to add additional information about your environment if you choose to print the worksheets.

If you are configuring for a failover group, fill in this portion of the worksheet:

Table 19.6. Sun Ray Server Failover Group Worksheet

Aspect or Variable	Default Value, Example, or (Other)	Your Primary Server Value	Your Secondary Server Value
Configuring the Sun Ray server hierarchy using utreplica (required for failover groups)	(Provide the start time)		
Primary Sun Ray server host name ⁽¹⁾	<i>primary-server</i>		
Secondary Sun Ray server host name ⁽¹⁾	<i>secondary-server</i>		

⁽¹⁾ These values are different for each Sun Ray server, even if that server is part of a failover group.

Table 19.7. First and Last Unit Address in a Failover Group

Server	First Unit Address	Last Unit Address
Primary	192.168.128.16	192.168.128.55
Secondary	192.168.128.56	192.168.128.95
Secondary	192.168.128.96	192.168.128.135
Secondary	192.168.128.136	192.168.128.175



Note

If you forget the address range, use [utadm -l](#) to list the addresses you specified or [utadm -p](#) to print them.

Chapter 20. Performance Tuning

Table of Contents

20.1. How to Gain Network Performance for Sun Ray 3 Series Clients	311
20.2. How to Improve Network Performance by Disabling CPU Binding (Oracle Solaris 11)	311
20.3. How to Improve Sun Ray Client Performance by Decreasing Buffering on the Network Switch (Oracle Solaris)	311
20.4. Improving Sun Ray Client Start-Up Time by Disabling Spanning Tree Protocol on the Network Switch	312
20.5. Applications	312
20.6. Tuning the Java Desktop System	313
20.7. Excessive Disk Swapping	313
20.8. Screensaver Resource Consumption	313
20.8.1. How to Disable Screensavers (Oracle Solaris 10)	313

This chapter provides tuning information for a Sun Ray environment.

20.1. How to Gain Network Performance for Sun Ray 3 Series Clients

To gain optimal network performance for Sun Ray 3 Series Clients, enable link-level flow control (IEEE 802.3 link pause) on the network switches to which the clients are connected.

20.2. How to Improve Network Performance by Disabling CPU Binding (Oracle Solaris 11)

If the performance of the Sun Ray Clients starts degrading after a few days, such as delayed character echoes or out-of-sync window and mouse actions, then this configuration update may help.

The following procedure disables CPU binding on the server.

1. Add the following line to the `/etc/system` file:

```
set mac:mac_cpu_binding_on=0
```

2. Reboot the system.

20.3. How to Improve Sun Ray Client Performance by Decreasing Buffering on the Network Switch (Oracle Solaris)

Some network switches do not work well with Sun Ray Clients when the server-side connection is configured to run at 1 Gbps. Because the Sun Ray Clients run at 100 Mbps and the data is sent from the X Windows server in periodic bursts, these switches are required to buffer a certain amount of data. This situation can happen even when the average data rate from the X server is well under 100 Mbps.

The X server is programmed in such a way that a certain allowed amount of data is sent at tick intervals. The original implementation had 50 ticks per second. The X server is allowed to send at a certain specific rate granted by the Sun Ray Client.

For example, if the Sun Ray Client's grant is 40 Mbps, it can send 5 MB per second in bursts that are sent every 1/50th of a second. This means that at each tick, the server can send 100 KB of data at a rate of 1

Gbps. This rate would cause a queue buildup in the switch of close to 100 KB, which would then drain out at 100 Mbps over the next 1/50th of a second.

The first action to mitigate this type of issue is to increase the number of ticks per second to 100 per second from 50. Thus, in the example above, the X server would send 50 KB every 10 ms rather than 100 KB every 20 ms. This setting would improve the situation considerably, but the problem would still remain. The 100 ticks per second rate was chosen because it corresponded to the normal resolution of the timer in Oracle Solaris and Oracle Linux.

To increase the number of ticks per second beyond 100, the operating system's timer must also be increased. For Oracle Solaris, use the following procedure.

Steps

1. Add the following line to the `/etc/system` file:

```
set hires_tick=1
```

2. Reboot the system.

The `hires_tick=1` setting increases the system timer resolution to 1000 ticks per second.

Because the X server code uses the system setting, the X server's bursts of data now use the same value, 1000 ticks = 1 second, that is, 1 tick = 1 ms. In the example, using the new tick duration results in the X server sending 5 KB of data every 1 ms.

Because the change to the tick duration decreases the amount of buffering required on the network switch, the performance of the Sun Ray Clients should improve.

20.4. Improving Sun Ray Client Start-Up Time by Disabling Spanning Tree Protocol on the Network Switch

The Sun Ray Clients are designed to power on and be fully operational in a very short time--typically less than 10 seconds.

Some network switches have initial configurations that can cause this start-up time to be considerably longer, often taking 30 seconds or longer to achieve a fully working state. Longer start-on times typically are due to the configuration of the Ethernet switch that implements capabilities not needed in the Sun Ray environment. The most common of these capabilities is enabling the spanning tree protocol, which is designed to detect and compensate for loops in the network.

In the Sun Ray environment, the spanning tree protocol should be disabled or deferred for ports connected directly to the Sun Ray Clients. Some manufacturers support a feature that immediately puts a port into the spanning tree forwarding mode. This feature is an acceptable alternative to disabling the spanning tree protocol on the port.

If the spanning tree protocol is disabled and the start-up time is still excessive, contact the switch manufacturer to determine if there are other features or proprietary protocols that might be interfering with the Sun Ray Client. Some switches might have features designed into the switch that cannot be changed; if this is the case, then it may not be possible to reduce the start-up time.

20.5. Applications

Some applications, such as intensive 3-D visual simulations, might run very slowly on a Sun Ray Client.

Applications that use double-buffering such as pseudo-stereo viewers and applications that use high-frequency dynamic color table flips on 8-bit visual displays might not show the proper result. Turn off antialiasing to save screen resources.

Install interactive applications such as web browsers and PC interoperability tools such as Oracle Secure Global Desktop (SGD) software on the Sun Ray server. The applications benefit from faster transport of commands to the Sun Ray Client's X server and network traffic is reduced.

If an application can be configured to use shared memory instead of DGA or OpenGL(R), using shared memory results in improved performance.

20.6. Tuning the Java Desktop System

To tune desktop performance, use solid backdrops and wireframe window moves.

For more instructions and recommendations, refer to the following information:

- [Java Desktop System documentation](#)
- [GNOME Performance Enhancement Tips for the Oracle Solaris Platform](#)
- [GNOME Performance Script for Oracle Solaris](#)

20.7. Excessive Disk Swapping

If the Sun Ray server does not have enough virtual memory available, the X Window server instance does not start and the user experiences a slow response. When virtual memory is insufficient, the Sun Ray server performs excessive disk swapping.

To determine whether the Sun Ray server is swapping to disk excessively, use the `vmstat` command:

```
# vmstat 5
```

If excessive swapping is occurring, the system might be undersized or overutilized.

The solution is to add more memory or increase the size of the swap partition.

20.8. Screensaver Resource Consumption

Many graphics-intensive screensaver programs consume large amounts of CPU, memory, and network bandwidth. To avoid excessive resource consumption on the Sun Ray servers, disable them.

20.8.1. How to Disable Screensavers (Oracle Solaris 10)

Remove the screensaver packages.

```
# pkgrm SUNWxscreensaver-hacks
# pkgrm SUNWxscreensaver-hacks-gl
```

If the `SUNWxscreensaver-hacks-gl` package is not removed successfully, remove the `gl` package and then remove the `SUNWxscreensaver-hacks-gl` package.

Appendix A. IPsec Support

Table of Contents

A.1. Overview	315
A.2. IKE Configuration	315
A.2.1. remote Directive	316
A.2.2. sainfo Directive	318
A.2.3. Example IKE Configuration Files	318
A.3. IPsec Configuration GUI Menu	319
A.4. IPsec Configuration Examples	319
A.4.1. Oracle Linux 5 Pre-Shared Key	320
A.4.2. Oracle Linux 5 Certificates	321
A.4.3. Oracle Linux 6 Pre-Shared Key	322
A.4.4. Oracle Linux 6 Certificates	323
A.4.5. Oracle Solaris Pre-Shared Key	324
A.4.6. Oracle Solaris Certificates	326
A.4.7. Sun Ray Client Configuration	327
A.4.8. IPsec Verification	328

This appendix provides information about IPsec support for IPv4 in Sun Ray Software.

A.1. Overview

Sun Ray Software supports Internet Protocol security (IPsec) to provide high quality, cryptographically-based security between Sun Ray Clients and Sun Ray servers. The security services offered with IPsec include access control, connectionless integrity, data origin authentication, protection against replays (a form of partial sequence integrity), confidentiality (encryption), and limited traffic flow confidentiality. These services are provided at the IP layer, offering protection for IP and upper-layer protocols.

The Sun Ray Software implementation of IPsec is incorporated into the Sun Ray Client firmware and is derived from an Open Source implementation provided at <http://ipsec-tools.sourceforge.net>. There are no IPsec implementation changes needed on the Sun Ray server, but the Sun Ray server requires the same general capabilities to configure IPsec, such as an IKE implementation and a tool to manage the security policy. The IPsec implementation is composed of three parts: the network stack implementation of IPsec itself that provides the security of the actual traffic; the Internet Key Exchange (IKE) implementation that manages the dynamic generation of keying material and authentication of the endpoints; and the policy management, through the Security Policy Database (SPD), that specifies which traffic should be protected and how.

After configuring and enabling IPsec on the Sun Ray server and the Sun Ray Client, the Sun Ray Client will negotiate a secure end-to-end IPsec tunnel with the Sun Ray server before interacting with Sun Ray services on the server. The Sun Ray Client will always be the initiator of a connection, so it does not have to respond to inbound connection requests. This type of negotiation is similar to the current IPsec VPN behavior, where IPsec is established with a VPN gateway before Sun Ray services are invoked. However, both IPsec implementations require different configurations.

A.2. IKE Configuration

The base configuration required for IPsec on the Sun Ray Client is the IKE configuration file, which is derived from the `racoon.conf` file. The IKE configuration file defines how to establish a secure connection between two hosts using the `racoon` daemon. For Sun Ray Software, only a subset of the

directives and statements in the IKE configuration file are required. The complete documentation for the IKE configuration file is available in the `racoon.conf` man page.

The IKE configuration file contains a set of directives that each consist of a keyword and a set of parameters. Some directives can be followed by a set of nested statements. An IKE configuration file must be stored in the firmware's security configuration repository at `/ike/default.conf`. The use of the `/ike` directory is consistent with the strategy that the directory containing a file indicates its type or format.

There are only two directives that are required for the Sun Ray IKE configuration file:

- `remote` - Specifies the parameters for IKE negotiations.
- `sainfo` - Specifies the parameters for protecting the actual IPsec traffic.

Normally, there can be multiple `remote` and `sainfo` directives, tagged with either a name, address, or the default `anonymous` keyword. The Sun Ray Software implementation allows for only one of each directive.

The following directives are not required, but they are supported to provide more advanced configuration:

- `padding` - Specifies the padding parameters for traffic.
- `timer` - Specifies the configuration timers.

Other parameters, such as the location of various ancillary files and ports, are fixed in value.

A.2.1. remote Directive

The `remote` directive specifies the parameters for IKE negotiations.

The following statements are supported. Specific notes and restrictions are provided where necessary.

- `ca_type` - Root certificate type (X509 only) and root certificate file name.
- `certificate_type` - Client certificate type (X509 only), private key file name, and certificate file name.
- `dpd` - Switch to enable Dead Peer Detection (DPD). Default is on.
- `dpd_delay` - Time between liveness requests. 0 disables checking. Default is 0.
- `dpd_maxfail` - If `dpd_delay` is set, this statement sets the maximum number of proof of liveness to request (without reply) before considering the peer is dead. The default value is 5.
- `dpd_retry` - If `dpd_delay` is set, this statement sets the delay (in seconds) to wait for a proof of liveness before considering it as failed and sending another request. The default value is 5.
- `exchange_mode` - Exchange mode to use as the IKE initiator. Values: `main`, `aggressive`, or `base`. The `aggressive` mode is not supported on Oracle Solaris.
- `ike_frag` - Switch to enable IKE fragmentation.
- `lifetime` - IKE lifetime proposed.
- `my_identifier` - Type and value of the IKE identifier for phase 1. The following identifier types are allowed:
 - `address` - IP address. This is the default, although this is not appropriate for Sun Ray Clients that get their addresses using DHCP.
 - `asn1dn` - ASN.1 distinguished name. This value is taken from the certificate Subject field if a value is not specified.

- `fqdn` - Fully-qualified domain name.
- `keyid` - An arbitrary string.
- `subnet` - IP subnet.
- `user_fqdn` - User fully-qualified name.
- `nat_traversal` - Switch to enable NAT traversal.
- `nonce_size` - Size of the nonces used in the IKE exchange. The default is 16 bytes.
- `peers_certfile` - Locally stored peer certificate type (X509 only) and certificate file name.
- `peers_identifier` - Type and value of the expected peer identifier. The following identifier types are allowed:
 - `address` - IP address. This is the default.
 - `asn1dn` - ASN.1 distinguished name. This value is taken from the certificate Subject field if a value is not specified.
 - `fqdn` - Fully-qualified domain name.
 - `keyid` - An arbitrary string.
 - `subnet` - IP subnet.
 - `user_fqdn` - User fully-qualified name.
- `proposal` - List of proposal statements. Only one proposal statement is allowed.
 - `authentication_method` - Specify the authentication method used. Values: `pre_shared_key` or `rsasig`.

The pre-shared key file is used when the authentication mode is `pre_shared_key`, and the file must be stored in the firmware's security configuration repository at `/preshared/keys` file. The pre-shared key file consists of lines containing pairs of ids and keys, separated by some number of blanks or tab characters. Keys starting with "0x" are interpreted as hexadecimal strings. Any referenced certificate files must be stored in the `/certs` directory, and public/private key pairs provided in files must be stored in the `/keys` directory.

- `dh_group` - Specify the group used for Diffie-Hellman exponentiation. Values: `modp768`, `modp1024`, `modp1536`, `modp2048`, `modp3072`, `modp4096`, `modp6144`, or the corresponding DH group number, 1, 2, 5, 14, 15, 16, 17, or 18.
- `encryption_algorithm` - Specify the encryption algorithm used for the phase 1 negotiation. Values: `aes`, `3des`, or `null`. `aes` may be followed by a key size of `128`, `192`, or `256`, separated by a space.
- `hash_algorithm` - Specify the hash algorithm used for phase 1 negotiation. Values: `md5` (deprecated), `sha1`, `sha256`, `sha384`, or `sha512`. Oracle Linux 5.8 and Oracle Linux 6.3 does not support the `sha384` or `sha512` hash algorithm.
- `lifetime` - Specify IKE lifetime.
- `remote_address` - Remote IP address of the other end of the connection.

- `proposal_check` - Type of proposal checking. Values: `claim`, `exact`, `obey`, or `strict`.
- `send_cert` - Switch to enable sending client certificate. Default is on.
- `send_cr` - Switch to enable sending certificate request. Default is on.
- `verify_cert` - Switch to verify the peer's certificate. Defaults is on.
- `verify_identifier` - Switch to enable verification of identity between ID and certificate. Default is off.

A.2.2. sainfo Directive

The `sainfo` directive is used to specify the security parameters for creating an IPsec Security Association (SA) used to protect associated traffic. For Sun Ray Software, only the Encapsulating Security Payload (ESP) is supported, and the Authentication Header (AH) protocol is not supported.

A full implementation of the Security Policy Database (SPD) for Sun Ray Software is not required, because the communication between the Sun Ray Client and other peers requires only a few switch selections, which have been incorporated into the IPsec configuration menu in the firmware Configuration GUI.

The following statements are supported. Specific notes and restrictions are provided where necessary.

- `authentication_algorithm` - Specify the comma-separated list of authentication algorithms. Values include hmac forms of the `hash_algorithm` values, such as `hmac_md5`, `hmac_sha1`, `hmac_sha256`, `hmac_sha384`, or `hmac_sha512`.
- `encryption_algorithm` - Specify the comma-separated set of encryption algorithms that can be used in a phase 2 proposal. Values: `aes` or `3des`. The `aes` value may be followed by a key size, for example, `aes 256`.
- `lifetime` - Define how long an IPsec SA will be used.
- `pfs_group` - Define the group used for Perfect Forward Secrecy (PFS) in phase 2. The same values are used as `dh_group`. If omitted, PFS is not used.
- `sha2_trunc` - Switch that sets the truncation of SHA-2 hashes to 96 bits, rather than the 128 specified in RFC 4868. This allows interoperation with some Oracle Linux systems that exhibit this behavior. This must be set on when using the `sha256` hash algorithm for Oracle Linux.

The proposals generated during the phase 2 negotiation consist of all of the possible combinations of `encryption_algorithm` and `authentication_algorithm`.

A.2.3. Example IKE Configuration Files

Here is an example of a Sun Ray IKE configuration file used to specify the connection between a Sun Ray Client with a fixed IP address (10.213.25.230) and a Sun Ray server (10.213.21.43) using a pre-shared key.

```
remote address 10.213.21.43 {
    my_identifier address 10.213.25.230;
    exchange_mode main;
    proposal {
        authentication_method pre_shared_key;
        encryption_algorithm aes;
        hash_algorithm sha1;
        dh_group 2;
    }
    proposal_check claim;
}
sainfo address 10.213.25.230 address 10.213.21.43 {
    lifetime time 12 hour;
```

```

encryption_algorithm aes;
authentication_algorithm hmac_shal;
}

```



Note

If you specify [main](#) for the [exchange_mode](#) statement, the identifiers for the IKE connection must be IP addresses when using pre-shared keys.

Here is another example of a Sun Ray IKE configuration file for certificate-based authentication

```

remote anonymous {
    exchange_mode main;
    my_identifier asnldn;
    ca_type x509 "cacert.pem";
    certificate_type x509 "mycert.pem" "mykey.pem";
    proposal {
        authentication_method rsasig;
        encryption_algorithm 3des;
        hash_algorithm md5;
        dh_group modp1024;
    }
    lifetime time 24 hour;
    proposal_check claim;
}
sainfo anonymous {
    authentication_algorithm hmac_shal;
    encryption_algorithm aes;
    lifetime time 8 hour;
}

```

A.3. IPsec Configuration GUI Menu

The firmware Configuration GUI provides the [Server/IPsec](#) menu to configure and enable IPsec. The top-level IPsec menu has the following entries:

- **IPsec Enable** - Switch to enable the use of IPsec.
- **Manage Policy** - Switches to enable or disable IPsec for individual types of access, for example, firmware and remote configuration file downloads.
- **Show Configuration** - Display the current IKE configuration file contents ([/ike/default.conf](#)).
- **Download Configuration** - Read a IKE configuration file from the [/tftpboot](#) directory and write it to [/ike/default.conf](#). Loading other files, which can include the IKE configuration file, must be done through the [Advanced > Download Configuration](#) menu.
- **Initialize Configuration** - Set the IKE configuration file to a known state, either as an empty file, or from a template.
- **Edit Configuration** - Read the existing IKE configuration file and edit individual statements in the file.
- **Manage Preshared Keys** - Display the current pre-shared keys, and create, delete, or modify entries.
- **Manage Public Keys** - Display the list of public keys and delete individual entries.

A.4. IPsec Configuration Examples

This sections provides examples show how to configure and enable IPsec on a Sun Ray server and a Sun Ray Client. For all of the examples, the following configuration information is used:

- Sun Ray Client - 10.25.198.65

- Sun Ray server - 10.213.21.168
- `sunray_ike.conf` - Sun Ray IKE configuration file
- `ikeload` - Remote configuration file
- `cacert.pem` - root certificate file
- `mycert.pem` - Certificate file
- `mykey.pem` - Secret key file

A.4.1. Oracle Linux 5 Pre-Shared Key

The following example shows how to configure IPsec using a pre-shared key on a Sun Ray server running Oracle Linux 5 and prepare an IKE configuration file for the Sun Ray Client.

1. Become superuser on the Sun Ray server.
2. Edit the `/etc/racoon/racoon.conf` file as follows:

```
path include "/etc/racoon";
path pre_shared_key "/etc/racoon/psk.txt";

remote anonymous {
    exchange_mode main;
    proposal {
        authentication_method pre_shared_key;
        encryption_algorithm 3des;
        hash_algorithm sha1;
        dh_group modp1024;
    }
    lifetime time 24 hour;
    proposal_check claim;
}
sainfo anonymous {
    authentication_algorithm hmac_sha1;
    encryption_algorithm 3des;
    lifetime time 8 hour;
    compression_algorithm deflate ;
}
```

3. Edit the `/etc/racoon/psk.txt` file to include the pre-shared key.

```
<ip-address_of_Sun_Ray_Client> <key>

10.25.198.65    0x12345678
```

4. Configure the SPD.

```
# setkey -c << EOF
spdadd 10.213.21.168 10.25.198.65 any -P out ipsec esp/transport//require;
spdadd 10.25.198.65 10.213.21.168 any -P in ipsec esp/transport//require;
```

Note that 10.213.21.168 is the Sun Ray server IP address and 10.25.198.65 is the Sun Ray Client IP address.

5. Create a `sunray_ike.conf` file for the Sun Ray Client with the following contents and save it to the `/tftpboot` directory.

```
remote anonymous {
    exchange_mode main;
    proposal {
```

```

        authentication_method pre_shared_key;
        encryption_algorithm 3des;
        hash_algorithm sha1;
        dh_group modp1024;
    }
    lifetime time 24 hour;
    proposal_check claim;
}
sainfo anonymous {
    authentication_algorithm hmac_sha1;
    encryption_algorithm 3des;
    lifetime time 8 hour;
}

```

6. Enable IPsec on the server if necessary.

```
# racoon
```

This manual step may not be necessary if IPsec is already enabled on the server. You can change the debug level by adding one or more `-d` options, such as `-ddd`.

A.4.2. Oracle Linux 5 Certificates

The following example shows how to configure IPsec using certificates on a Sun Ray server running Oracle Linux 5 and prepare an IKE configuration file for the Sun Ray Client.

1. Become superuser on the Sun Ray server.
2. Copy the `cacert.pem`, `mycert.pem`, and `mykey.pem` files to the `/etc/racoon/certs` and `/tftpbboot` directories.
3. Edit the `/etc/racoon/racoon.conf` file as follows:

```

path include "/etc/racoon";
path certificate "/etc/racoon/certs";

remote anonymous {
    exchange_mode main;
    generate_policy on;
    passive on;
    ca_type x509 "cacert.pem";
    certificate_type x509 "mycert.pem" "mykey.pem";
    my_identifier asn1dn;
    peers_identifier asn1dn;
    proposal_check claim;
    lifetime time 24 hour;
    proposal {
        encryption_algorithm 3des;
        hash_algorithm md5;
        authentication_method rsasig;
        dh_group modp1024;
    }
}

sainfo anonymous {
    pfs_group modp1024;
    encryption_algorithm 3des;
    authentication_algorithm hmac_sha1;
    lifetime time 8 hour;
    compression_algorithm deflate;
}

```

4. Create a `sunray_ike.conf` file for the Sun Ray Client with the following contents and save it to the `/tftpbboot` directory.

```

remote anonymous {
    exchange_mode main;
    my_identifier asn1dn;
    ca_type x509 "cacert.pem";
    certificate_type x509 "mycert.pem" "mykey.pem";
    proposal {
        authentication_method rsasig;
        encryption_algorithm 3des;
        hash_algorithm md5;
        dh_group modp1024;
    }
    lifetime time 24 hour;
    proposal_check claim;
}
sainfo anonymous {
    pfs_group modp1024;
    authentication_algorithm hmac_shal;
    encryption_algorithm 3des;
    lifetime time 8 hour;
}

```

5. Create a remote configuration file named `ikeload` with the following contents and save it to the `/tftpbboot` directory.

```

/certs/cacert.pem=cacert.pem
/keys/mykey.pem=mykey.pem
/certs/mycert.pem=mycert.pem
/ike/default.conf=sunray_ike.conf

```

6. Enable IPsec on the server if necessary.

```
# racoon
```

This manual step may not be necessary if IPsec is already enabled on the server. You can change the debug level by adding one or more `-d` options, such as `-ddd`.

A.4.3. Oracle Linux 6 Pre-Shared Key

The following example shows how to configure IPsec using a pre-shared key on a Sun Ray server running Oracle Linux 6 and prepare an IKE configuration file for the Sun Ray Client..

1. Become superuser on the Sun Ray server.
2. If not already installed, install the `openswan-2.6.32-16.el6.x86_64.rpm` RPM.
3. Uncomment the following line in the `/etc/ipsec.conf` file:

```
include /etc/ipsec.d/*.conf
```

4. Make sure the `/etc/ipsec.secrets` file contains only the following line:

```
include /etc/ipsec.d/*.secrets
```

5. Create the `/etc/ipsec.d/shared.conf` file with the following contents, which includes the Sun Ray server and the Sun Ray Client IP addresses for the `left` and `right` entries, respectively:

```

conn new
    left=10.213.21.168
    right=10.25.198.65
    authby=secret
    type=transport
    ike=3des-md5;modp1024
    esp=3des-md5

```



```
keyexchange=ike
pfs=no
rekey=no
aggrmode=no
phase2=esp
salifetime=8h
auto=add
```

6. Create the `/etc/ipsec.d/shared.secrets` file with the following contents, which includes an entry containing the Sun Ray server and Sun Ray Client IP addresses and the pre-shared key:

```
10.213.21.168 10.25.198.65: PSK "12345678"
```

7. Create a `sunray_ike.conf` file for the Sun Ray Client with the following contents and save it to the `/tftpbboot` directory.

```
remote anonymous {
    exchange_mode main;
    proposal {
        authentication_method pre_shared_key;
        encryption_algorithm 3des;
        hash_algorithm sha1;
        dh_group modp1024;
    }
    lifetime time 24 hour;
    proposal_check claim;
}
sainfo anonymous {
    authentication_algorithm hmac_sha1;
    encryption_algorithm 3des;
    lifetime time 8 hour;
}
```

8. Start the IPsec services.

```
# /etc/init.d/ipsec start
```

A.4.4. Oracle Linux 6 Certificates

The following example shows how to configure IPsec using certificates on a Sun Ray server running Oracle Linux 6 and prepare an IKE configuration file for the Sun Ray Client.

1. Become superuser on the Sun Ray server.
2. If not already installed, install the `openswan-2.6.32-16.el6.x86_64.rpm` RPM.
3. Uncomment the following line in the `/etc/ipsec.conf` file:

```
include /etc/ipsec.d/*.conf
```

4. Make sure the `/etc/ipsec.secrets` file contains only the following line:

```
include /etc/ipsec.d/*.secrets
```

5. Create the `/etc/ipsec.d/certs.conf` file with the following contents:

```
conn new1
    left=10.213.21.168
    right=%any
    leftcert="server_certificate"
    rightcert="client_certificate"
    leftid=%fromcert
```

```

rightid=%fromcert
authby=rsasig
leftrsasigkey=%cert
type=transport
ike=aes-sha2_256;modp1024
phase2alg=aes-sha2_256
keyexchange=ike
keyingtries=3
pfs=no
rekey=no
aggrmode=no
phase2=esp
salifetime=8h
auto=add

```

The `right=%any` entry enables any client to connect with the proper certificate.

6. Create the `/etc/ipsec.d/certs.secrets` file with the following contents, which includes the Sun Ray server:

```
%any : RSA 10.213.21.168
```

7. Create a `sunray_ike.conf` file for the Sun Ray Client with the following contents and save it to the `/tftboot` directory.

```

remote anonymous {
    exchange_mode main;
    my_identifier asn1dn;
    ca_type x509 "cacert.pem";
    certificate_type x509 "mycert.pem" "mykey.pem";
    proposal {
        authentication_method rsasig;
        encryption_algorithm 3des;
        hash_algorithm sha1;
        dh_group modp1024;
    }
    lifetime time 24 hour;
    proposal_check claim;
}
sainfo anonymous {
    authentication_algorithm hmac_sha1;
    encryption_algorithm 3des;
    lifetime time 8 hour;
}

```

8. Create a remote configuration file named `ikeload` with the following contents and save it to the `/tftboot` directory.

```

/certs/cacert.pem=cacert.pem
/keys/mykey.pem=mykey.pem
/certs/mycert.pem=mycert.pem
/ike/default.conf=sunray_ike.conf

```

9. Start the IPsec services.

```
# /etc/init.d/ipsec start
```

A.4.5. Oracle Solaris Pre-Shared Key

The following example shows how to configure IPsec using a pre-shared key on a Sun Ray server running Oracle Solaris 10 or Oracle Solaris 11 and prepare an IKE configuration file for the Sun Ray Client.

1. Become superuser on the Sun Ray server.

2. Edit the `/etc/inet/ike/config` file as follows:

```
p1_lifetime_secs 86400
p1_nonce_len 16

p2_lifetime_secs 28800

## Parameters that may also show up in rules.

p1_xform { auth_method preshared oakley_group 2 auth_alg sha1 encr_alg aes }

p2_pfs 0

### Now some rules...

{
    label "SRSS Rule"

    # Use whatever "host" (e.g. IP address) identity is appropriate
    local_addr 0.0.0.0/0
    remote_addr 0.0.0.0/0

    p1_xform
    { auth_method preshared oakley_group 2 auth_alg sha encr_alg aes }

    p2_pfs 0
}
```

3. Edit the `/etc/inet/secret/ike.preshared` file to include the pre-shared key.

```
{
    localidtype      IP
    localid          10.213.21.168
    remoteidtype     IP
    remoteid         10.25.198.65
    key              12345678
}
```

4. Configure the IPsec policy by adding the following line to the `/etc/inet/ipsecinit.conf` file:

```
{ laddr 10.213.21.168 raddr 10.25.198.65 } ipsec {encr_algs aes encr_auth_algs sha1}
```

5. Create a `sunray_ike.conf` file for the Sun Ray Client with the following contents and save it to the `/tftpboot` directory.

```
remote anonymous {
    exchange_mode main;
    proposal {
        authentication_method pre_shared_key;
        encryption_algorithm aes;
        hash_algorithm sha1;
        dh_group 2;
    }
    lifetime time 24 hour;
    proposal_check claim;
}
sainfo anonymous {
    authentication_algorithm hmac_sha1;
    encryption_algorithm aes;
    lifetime time 8 hour;
}
```

6. Enable IPsec on the server.

```
# svcadm restart svc:/network/ipsec/ipsecalgs:default
# svcadm restart svc:/network/ipsec/policy:default
```

```
# /usr/lib/inet/in.iked
```

You can use the `svcs | grep ipsec` command to verify that IPsec is enabled. You can use the `-d` option of the `in.iked` command to keep it in the foreground and produce debugging output.

A.4.6. Oracle Solaris Certificates

The following example shows how to configure IPsec using certificates on a Sun Ray server running Oracle Solaris 10 or Oracle Solaris 11 and prepare an IKE configuration file for the Sun Ray Client..

1. Become superuser on the Sun Ray server.
2. Copy the `cacert.pem`, `mycert.pem`, and `mykey.pem` files to the `/etc/racoon/certs` and `/tftpbboot` directories.
3. Edit the `/etc/inet/ike/config` file as follows:

```
####

cert_root    "C=US, L=Redwood Shores, ST=CA, O=Company, OU=Sun Ray,
              CN=First Last, MAILTO=first.last@company.com"

ignore_crls

p1_lifetime_secs 86400
p1_nonce_len 16

p2_lifetime_secs 28800

p1_xform { auth_method rsa_sig oakley_group 2 auth_alg sha encr_alg 3des }

p2_pfs 0

{
    label "SRSS Rule"

    local_id_type dn
    local_id "C=US, L=Redwood Shores, ST=CA, O=Company, OU=Sun Ray, CN=server-fqdn"
    remote_id ""

    local_addr 0.0.0.0/0
    remote_addr 0.0.0.0/0

    p1_xform
    { auth_method rsa_sig oakley_group 2 auth_alg md5 encr_alg 3des }

    p2_pfs 0
}

####
```

4. Configure the IPsec policy by adding the following line to the `/etc/inet/ipsecinit.conf` file:

```
{ laddr 10.213.21.168 raddr 10.25.198.65 } ipsec {encr_algs 3des encr_auth_algs sha1}
```

5. Create a `sunray_ike.conf` file for the Sun Ray Client with the following contents and save it to the `/tftpbboot` directory.

```
remote anonymous {
    exchange_mode main;
    my_identifier asnldn;
    ca_type x509 "cacert.pem";
    certificate_type x509 "mycert.pem" "mykey.pem";
    proposal {
```

```

        authentication_method rsasig;
        encryption_algorithm 3des;
        hash_algorithm md5;
        dh_group 2;
    }
    lifetime time 24 hour;
    proposal_check claim;
}
sainfo anonymous {
    pfs_group modp1024;
    authentication_algorithm hmac_shal;
    encryption_algorithm 3des;
    lifetime time 8 hour;
}

```

6. Create a remote configuration file named `ikeload` with the following contents and save it to the `/tftboot` directory.

```

/certs/cacert.pem=cacert.pem
/keys/mykey.pem=mykey.pem
/certs/mycert.pem=mycert.pem
/ike/default.conf=sunray_ike.conf

```

7. Enable IPsec on the server.

```

# svcadm restart svc:/network/ipsec/ipsecalgs:default
# svcadm restart svc:/network/ipsec/policy:default
# /usr/lib/inet/in.iked

```

You can use the `svcs | grep ipsec` command to verify that IPsec is enabled. You can use the `-d` option of the `in.iked` command to keep it in the foreground and produce debugging output.

A.4.7. Sun Ray Client Configuration

Once you configure IPsec on the Sun Ray server, including the adding the appropriate Sun Ray IKE configuration file and certificates to the `/tftboot` directory, there are only a few steps remaining to configure IPsec on the Sun Ray Client using the Configuration GUI. The following steps continue the previous Sun Ray server configuration examples.

1. Open the Configuration GUI on the Sun Ray Client.

See [Section 14.5.2, "Configuration GUI Menu Descriptions"](#) for details.

2. Load the configuration files on Sun Ray Client from the server's `/tftboot` directory:
 - a. If you have only a Sun Ray IKE configuration file to load, choose [Server/IPsec > Download Configuration](#) and specify the server and the IKE configuration file. For the pre-shared examples in this section, you would enter `10.213.21.168/sunray_ike.conf` to populate the `/ike/default.conf` file in the Sun Ray Client's firmware.
 - b. If you are using a remote configuration file to load a number of files, choose [Advanced > Download Configuration](#) and enter the server and the remote configuration file. For the certificate examples in this section, you would enter `10.213.21.168/ikeload` to populate the IKE configuration file and the certificate files in the Sun Ray Client's firmware.
3. Choose [Server/IPsec](#).
4. For the pre-shared key examples in this section, choose [Manage Preshared Keys](#) to create the pre-shared key:

```

10.25.198.65    0x12345678

```

You can also use the remote configuration file to load a pre-shared key.

5. Choose [IPsec Enable](#) and enable IPsec.
6. Exit the Configuration GUI.

A.4.8. IPsec Verification

After configuring IPsec on the Sun Ray server and Sun Ray Client, you can verify if IPsec is working by rebooting the Sun Ray Client with the OSD icons enabled. If the IPsec OSD network status icons is displayed with the up arrow, IPsec should be working.

To verify if the traffic is being encrypted between the server and the Sun Ray, use a network monitoring tool (for example, [snoop](#) or [tcpdump](#)) and confirm that the packets seen are using the ESP protocol.

Appendix B. Admin GUI Help

Table of Contents

B.1. Servers Tab	329
B.1.1. Server Details	330
B.2. Sessions Tab	331
B.3. Desktop Units Tab	333
B.3.1. Desktop Units Properties	334
B.4. Tokens Tab	335
B.4.1. Token Properties	337
B.5. Advanced Tab	338
B.5.1. Security	338
B.5.2. System Policy	339
B.5.3. Kiosk Mode	341
B.5.4. Card Probe Order	342
B.5.5. Data Store Password	342
B.6. Log Files Tab	343

This appendix provides reference help for the Sun Ray Software Administration GUI, or Admin GUI. See [Section 4.2, “Administration Tool \(Admin GUI\)”](#) for more information.

B.1. Servers Tab

This page lists all the Sun Ray servers that are in the same failover group, which means they share the same group signature.

To ensure uninterrupted service, several Sun Ray servers can be configured as a failover group. Servers in a group authenticate (or learn to trust) one another by using a common group signature, a key used to sign messages sent between servers in the group. See [Chapter 6, Failover Groups](#) for more information.

Actions

- Display the high-level details for all the Sun Ray servers in the same failover group, including the servers' availability to the failover group.
- Display more details on a server by clicking on the name of the server. See [Section B.1.1, “Server Details”](#) for details.
- Perform a warm or cold restart on one or more servers.

Restart Buttons

Some configuration changes do not take effect until the Sun Ray services on each server are restarted. If a change requires Sun Ray services to be restarted, a notification message is displayed in the upper left-hand side of the Admin GUI.

To perform a group-wide restart, select all servers in the servers table (either individually or with the Select All button) before clicking the Warm Restart or Cold Restart button.

- **Warm Restart** - Existing Sun Ray sessions are preserved. Use this option if a minor configuration change was made. With minor changes, it is not necessary to terminate existing sessions.
- **Cold Restart** - All existing Sun Ray sessions are terminated. Use this option if a significant change has been made.

Servers Table

- **Name** - Domain Name System (DNS) name of the server. Click on the name to display the server details.
- **IP Address** - The primary IP address of the server.
- **Mode** - Specifies if the server is Online or Offline.
 - **Online** - The server participates in the normal session creation process controlled by the load balancing algorithm in the failover group.
 - **Offline** - The server does not participate in load balancing (the load balancing algorithm does not select this server for new sessions), although sessions can still be created on it, either explicitly, through the use of the `utswitch` or `utselect` command, or implicitly, if all other servers are down.
- **LAN Connections** - Specifies if the server is configured as a shared network and can accept client connections from the shared network.
- **Start Time** - Time of the last cold or warm restart on the sever.

Related Commands

- `utgstatus`
- `utstart`
- `utstop`

B.1.1. Server Details

This page provides details on a specific Sun Ray server.

Actions

- Display specific details about a Sun Ray server, including the Sun Ray Software version installed, status of connected clients and sessions, and system information.
- Perform a warm or cold restart on the Sun Ray server.

Restart Buttons

See [Section B.1, “Servers Tab”](#) for details.

General Section

- **Sun Ray Software** - The Sun Ray Software version installed on the server. Click [View Installed Sun Ray Packages](#) to display the list of packages or RPMs.
- **Primary IP Address** - The primary IP address of the server.
- **Type** - [need description] Standalone
- **Mode** - Specifies if the server is Online or Offline. See [Section B.1, “Servers Tab”](#) for details.
- **LAN Connections** - Specifies if the server is configured as a shared network and can accept client connections from the shared network.

- **Connectivity** - Click [View Network Status](#) to display the network status for all the trusted servers in the failover group from the perspective of the selected server.
- **Start Time** - Time of the last cold or warm restart on the sever.

Status Section

- **Connected Desktop Units** - Number of Sun Ray Clients and Oracle Virtual Desktop Clients currently connected to the Sun Ray server. If there are connected clients, click [View Desktop Units](#) to display more details about the clients, which is described in [Section B.3.1, "Desktop Units Properties"](#).
- **Total User Sessions** - Number of user sessions currently running on the Sun Ray server. If there are user sessions currently running, click [View Session Details](#) to display more details about the sessions, which is described in [Section B.2, "Sessions Tab"](#).
- **Logged in Users** - Number of users currently logged in to a session.
- **Login Greeter/Idle Sessions** - Number of login greeter or idle sessions currently running. If there are login greeter or idle sessions running, click [View Session Details](#) to display more details about the sessions, which is described in [Section B.2, "Sessions Tab"](#).

System Information Section

- **Operating System** - The operating system version installed on the Sun Ray server.
- **Disk Space** - A table displaying all the server's file systems and their usage.

Related Commands

- [utgstatus](#)
- [utrelease](#)
- [utstart](#)
- [utstop](#)

B.2. Sessions Tab

This page lists the Sun Ray sessions, including the user sessions and login greeter/idle sessions. Sun Ray sessions are groups of services or applications that are associated with an authentication token. They reside on a Sun Ray server and can be directed to any client. See [Chapter 7, Sessions and Tokens](#) for more information.

Actions

- Filter the list of sessions by using the Search field, such as sessions running on all Sun Ray servers or a single server.
- Display the sessions on all the servers or a specific server, sorted by user sessions and idle sessions.
- Display details about the server or client associated with the session.
- Terminate one or more sessions.

Search Field

The search field enables you to filter what sessions are displayed in the Sessions table. The drop-down menu enables you to filter the sessions on all the Sun Ray servers in the failover group or a single server.

By default, a wildcard (*) is in the search field to display all the sessions based on the selected filter. You can also specify a session token name or partial name in the field to further filter the list of sessions that are displayed. After you specify a new search string, you can always click [Reset Search Criteria](#) to reset the search back to the default.

Terminate Button

To terminate one or more sessions, select the sessions in the sessions list (either individually or with the Select All button) and then click Terminate.

Sessions Table

- **User Sessions** - List of the user sessions based on the current search filter. These are sessions with UNIX users logged in, also known as non-idle sessions. Users may start up additional applications from within the sessions, thus potentially consuming a lot of system resources. User sessions are therefore of more interest to administrators than idle sessions. To free system resources, monitor the number of long-running disconnected user sessions and, when appropriate, terminate sessions that are no longer in use.
- **Login Greeter/Idle Sessions** - List of sessions at the login greeter stage based on the current search filter. These sessions typically display only a login screen (or login greeter such as [dtlogin](#) or [gdm](#)) where no user has been logged in yet. The lifetime of these sessions is controlled by the Sun Ray system. For example, disconnected idle sessions are automatically terminated (reaped off) by the system after a specific time interval.
- **Token** - The token ID for the session.
- **Owner** - The owner of the session, which is set if the token for the session is registered.
- **Unix ID** - The Unix ID of the session owner.
- **Server** - The Sun Ray server on which the session is currently running. Click on the server name for more details on the server. See [Section B.1, "Servers Tab"](#) for details.
- **Display** - The number of the X server display for the session.
- **Status** - Specifies if the session is [Connected](#) or [Disconnected](#).
 - **Connected** - Sessions with a connected status are currently displayed on a client. The session is automatically disconnected when the user removes the smart card or explicitly switches the client to a different session, for example, with the [utswitch](#) or [utselect](#) commands.
 - **Disconnected** - Sessions still running on a server but are not connected to a client and, consequently, are not displayed. However, a user can reconnect to a disconnected session, such as inserting a smart card containing the appropriate token into the card reader on a client. This changes the session's state to connected and causes it to be displayed on that client.
- **Desktop Unit** - The client ID (full version) of the client currently connected to the session. Examples include [IEEE802.080020b5ca55](#) for a Sun Ray Client and [MD5.d8b3a4eb29497e0c6fbb0f2a810267f5](#) for an Oracle Virtual Desktop Client. Click the client ID for more details on the client. See [Section B.3, "Desktop Units Tab"](#) for details.

Related Commands

- [utdetach](#)
- [utsession](#)

B.3. Desktop Units Tab

This page lists all the registered desktop units (DTUs), which include the Sun Ray Clients and Oracle Virtual Desktop Clients. The term desktop client or client is also used for desktop unit or DTU.

Whenever a client connects for the first time to a Sun Ray server, it is automatically registered in the Sun Ray data store. (You can add new clients to the data store manually if you want to register any information before the first connection time.) It is also possible to store the client's location or other information that might help you to manage the clients in your organization.

Each Sun Ray Client also has a public-private key pair stored in the Sun Ray data store used for client authentication. Each stored key is annotated with a status, which indicates whether the authenticity of the key for the client with a given MAC address has been verified and confirmed by the server. Keys for Oracle Virtual Desktop Clients are not stored with the data store entry for the client, as the client identifier is directly related to the key.

If client authentication and hard security are enabled as part of your Sun Ray Software policy, Sun Ray Clients cannot be used until its client key is confirmed. The client key for an Oracle Virtual Desktop Client is automatically confirmed.

See [Chapter 11, Client-Server Security](#) for more information.

Actions

- Filter the list of desktop units displayed by using the Search field. You can filter the list from the following groupings: desktop units, connected desktop units, token readers, and multihead groups.
- Add a new desktop unit to the list. A client is automatically added to the Sun Ray data store when a client first connects, but you can preregister information for a client.
- Delete one or more desktop units. This removes all the information for the client from the Sun Ray data store.
- Display details about the desktop units.
- Edit details about a desktop unit by clicking on the client identifier. See [Section B.3.1, "Desktop Units Properties"](#) for details.

Search Field

The search field enables you to filter what is displayed in the Desktop Units table. The drop-down menu enables you to display different types of desktop units, including all desktop units, connected desktop units, token readers, or multihead groups. By default, a wildcard (*) is in the search field to display all the desktop units that match the current type. You can also specify the client IDs or partial IDs in the search field to further filter the list of desktop units that are displayed. After you specify a new search string, you can always click [Reset Search Criteria](#) to reset the search back to the default.

New and Delete Buttons

If nothing is selected in the table, the New button is available. You can use the New button to preregister information about clients to be used. If you select one or more existing clients in the table (either individually or with the Select All button), you can use the Delete button to delete them from the Sun Ray data store.

Desktop Units Table

- **Identifier** - The client ID (short version) of the client connected to the session. Examples include [080020b5ca55](#) for a Sun Ray Client and [d8b3a4eb29497e0c6fbb0f2a810267f5](#) for an Oracle

Virtual Desktop Client. Click the client ID for more details on the client. See [Section B.3, “Desktop Units Tab”](#) for details.

- **Location** - The location of the desktop unit (user defined).
- **Other Information** - General information about the desktop unit (user defined).
- **Client Key Status** - Specifies the status of the client authentication.
 - **Automatic** - The client key for an Oracle Virtual Desktop Client is automatically confirmed.
 - **Confirmed** - The client key has been confirmed by the server.
 - **Conflict** - If different keys are used with the same client identifier and none of the keys have been confirmed to be authentic, then this may indicate a security incident and is marked as a conflict. In this case, connection attempts from this client are rejected until one of the keys is confirmed to be valid.
 - **Unconfirmed** - The client key has not been confirmed by the server. If hard security is set, the client will not be able to connect.

Related Commands

- [utdesktop](#)

B.3.1. Desktop Units Properties

This page provides details on a desktop unit and enables you to edit its properties.

Actions

- Display more details about a desktop unit.
- Edit details about a desktop unit, including location and other information.
- Confirm or delete the client keys for the desktop unit.
- Configure a Sun Ray Client to be a token reader.

Edit Button

The Edit button enables you to edit the location and other information for a desktop unit. You can also configure a Sun Ray Client to be a token reader.

General Section

- **Location** - The location of the desktop unit (user defined).
- **Other Information** - General information about the desktop unit (user defined).
- **Model** - The model name for the desktop unit. The model names are the same names used in the server's TFTP home directory. The [SunRayS1](#) model name specifies an Oracle Virtual Desktop Client.
- **Firmware** - If connected, displays the Sun Ray Operating Software version currently installed on the Sun Ray Client.

Status Section

- **Current Token** - If connected, displays the current token used for the connection. Click [View Token Details](#) for more details.

- **Unix User** - If connected, displays the Unix user ID logged in to the desktop unit. Click [Terminate Session](#) or [Terminate Idle Session](#) to terminate the session on the desktop unit.
- **Server** - If connected, displays the Sun Ray server to which the desktop unit is connected. Click [View Server Details](#) for more details (see [Section B.1.1, “Server Details”](#) for details).
- **Status** - Specifies if the session is [Connected](#) or [Disconnected](#).
 - **Connected** - Sessions with a connected status are currently displayed on a client. The session is automatically disconnected when the user removes the smart card or explicitly switches the client to a different session, for example, with the [utswitch](#) or [utselect](#) commands.
 - **Disconnected** - Sessions still running on a server but are not connected to a client and, consequently, are not displayed. However, a user can reconnect to a disconnected session, such as inserting a smart card containing the appropriate token into the card reader on a client. This changes the session's state to connected and causes it to be displayed on that client.
- **Client Keys** - List the client keys for the Sun Ray Client. Once selected, you can confirm or delete a client key.

Advanced

- **Token Reader** - Specifies if the Sun Ray Client is a token reader. Click [Edit](#) to configure the desktop unit as a token reader. Sun Ray Clients configured as token readers do not support hotdesking. They are used for registration of new tokens.

Related Commands

- [utdesktop](#)
- [utkeyadm](#)
- [utreader](#)

B.4. Tokens Tab

Tokens are authentication keys used to associate a session with a user. A token is a string that consist of a token type and an identifier. If a user inserts a smart card into a client, the card's type and identifier are used as the token (for example [mondex.9998007668077709](#)). If the user is not using a smart card, the token type [pseudo](#) and the client's identifier (MAC address) are supplied as the token (for example [pseudo.080020861234](#)).

You can register smart card tokens and pseudo-tokens in the Sun Ray data store to assign them to specific users (also known as token owners). You can store the owner's name as well as any other information that helps you to manage tokens in your organization. You can also register alias tokens to enable users to access the same session with multiple tokens. For example, if a user loses a smart card, you can register a new smart card as a replacement. This will be an alias token.

If kiosk mode is configured, you can also specify, for each token, whether the user should be directed to a regular (non-kiosk) session or a kiosk session when the token is inserted. This enables you to override the group-wide kiosk mode setting specified on the System Policy page.

See [Chapter 7, Sessions and Tokens](#) for more information.

Actions

- Filter the list of tokens displayed by using the Search field. You can filter the list from the following groupings: registered tokens or currently used tokens.

- Display details about the tokens.
- Pre-register smart card tokens and pseudo-tokens.
- Delete one or more tokens.
- Enable or disable session access for a specific token.
- Assign a regular (non-kiosk) session or kiosk session to a token.
- Edit a token, including registering it, by clicking on the token identifier. See [Section B.4.1, “Token Properties”](#) for details.

Search Field

The search field enables you to filter what tokens are displayed in the Tokens table. The drop-down menu enables you to filter the tokens by registered tokens or currently used tokens. By default, a wildcard (*) is in the search field to display all the tokens for the specific filter. You can also specify a token ID or partial ID in the field to further filter the list of tokens that are displayed. After you specify a new search string, you can always click [Reset Search Criteria](#) to reset the search back to the default.

New and Delete Buttons

If nothing is selected in the table, the New button is available. You can use the New button to preregister a token, which enables you to specify the identifier for a token and its owner. You can also specify the session type used for the token, which overrides the current policy settings.

If you select one or more existing tokens in the table (either individually or with the Select All button), you can use the Delete button to delete them.

Enable and Disable Buttons

If you have the system policy set to accept access from only registered tokens, you can then enable or disable session access at the token level. For example, if you disable a token, the smart card or client associated with that token will not be able to access a session.

Session Type Actions

If you select one or more existing tokens in the table (either individually or with the Select All button), you can use the Session Type Actions menu to specify the session type used for the token, which overrides the current policy settings. This is a good way to change the session type for a group of tokens.

Token Table

The Token table content changes based on which filter is selected, either Registered Tokens or Currently Used Tokens. The buttons are available only when the Registered Token filter is used.

- **Token** - The token ID. Examples include [mondex.9998007668077709](#) for a smart card token or [pseudo.080020861234](#) for a pseudo token. Click the token ID for more details on the token. See [Section B.4.1, “Token Properties”](#) for details.
- **Status** - Enabled or Disabled. If you have the system policy set to accept access from only registered tokens, you can then enable or disable access at the token level.
- **Session Type** - The session type defined for the token, which overrides the system policy.
 - **Default** - Use the default system policy.

- **Regular** - Use the regular (non-kiosk) mode session, which is the Sun Ray server's operating system.
- **Kiosk** - Use the configured kiosk session type, which is listed in the Kiosk tab.
- **Owner** - The owner registered to the token (user defined).
- **Other Information** - Additional information provided for the token (user defined).
- **Server** - The Sun Ray server to which the token is currently assigned.
- **Desktop Unit** - The client to which the token is currently assigned.

Related Commands

- `utsession`
- `utuser`

B.4.1. Token Properties

This page provides details on a token and enables you to edit its properties.

Actions

- Display more details about a token.
- Edit details about a token, including owner, other information, and session type. Adding an owner registers the token.
- Add alias tokens.

Edit Button

The Edit button enables you to edit the owner, other information, status, and session type for the token.

General Section

- **Owner** - The owner registered to the token (user defined).
- **Other Information** - General information about the token (user defined).
- **Status** - Enabled or disabled. If you have the system policy set to accept access from only registered tokens, you can then enable or disable access at the token level.

Advanced Section

- **Sessions** - Specifies the number of running sessions associated with the token. If there are one or more sessions, you can click View Session Details to view the list of sessions.
- **Session Type** - Specifies the session type assigned to the token.
- **Alias Tokens** - Enables you to create alias tokens for the current token.

Related Commands

- `utsession`

- `utuser`

B.5. Advanced Tab

This tab provides various subpages to administer the following features:

- **Security** - Configure encryption and authentication between client and server and to enable or disable devices globally.
- **System Policy** - Configure group-wide Sun Ray Software policies.
- **Kiosk Mode** - Configure the kiosk session type used for kiosk sessions.
- **Card Probe Order** - Display all the smart card types configured in the Sun Ray data store and set the smart card prob order.
- **Data Store Password** - Change the password of the administrative user.

B.5.1. Security

This page enables you to configure the security policies for the Sun Ray server. See [Chapter 11, Client-Server Security](#) for more information.

Actions

- Configure encryption and server authentication.
- Configure client authentication.
- Enable or disable access to client devices attached to the Sun Ray Clients.
- Enable or disable the clipboard on Oracle Virtual Desktop Clients.

Encryption and Server Authentication Section

- **Upstream Encryption** - Select to enable encryption from the client to the Sun Ray server.
- **Downstream Encryption** - Select to enable encryption from the Sun Ray server to the client.
- **Server Authentication** - Select to force a server to be authenticated before providing a session to a client.
- **Security Mode** - Choose the security mode for the encryption and server authentication:
 - **Soft** - Ensures that connection requests are granted even for Sun Ray Clients that don't support the configured security requirements. If security requirements cannot be met, the session is granted but not secure.
 - **Hard** - Ensures that every session is secure. If security requirements cannot be met, the session is refused.



Note

Security mode settings don't apply to Oracle Virtual Desktop Clients. Oracle Virtual Desktop Clients will always be treated as if hard security mode for encryption or authentication is in effect.

Client Authentication Section

- **Client Authentication** - Select to force a client to be authenticated before obtaining a session. A Sun Ray Client whose key has not been confirmed as valid for the given Sun Ray Client will still be allowed access to Sun Ray sessions by default, unless there is a conflict when the client ID (the MAC address) is used with multiple keys. To force client key confirmation, see the See [Section B.5.2, "System Policy"](#) for details.
- **Security Mode** - Choose the security mode for client authentication:
 - **Soft** - Ensures that connection requests are granted even for Sun Ray Clients that don't support the configured security requirements. If security requirements cannot be met, the session is granted but not secure.
 - **Hard** - Ensures that every session is secure. If security requirements cannot be met, the session is refused.



Note

Security mode settings don't apply to Oracle Virtual Desktop Clients. Oracle Virtual Desktop Clients will always be treated as if hard security mode for authentication is in effect.

Devices Section

- **Internal Serial Port** - Select to enable access to the serial port on the Sun Ray Clients.
- **Internal Smart Card Reader** - Select to enable access to the smart card readers on the Sun Ray Clients. Choose the smart card protocol to use, either [scbus v1](#) or [scbus v2](#). Choose [scbus v2](#) unless you are managing Sun Ray Clients running Sun Ray Software 5.2 firmware or earlier.
- **USB Port** - Select to enable access to the USB ports on the Sun Ray Clients.
- **Oracle Virtual Desktop Client Clipboard** - Select to enable copy and paste text between an application running in an Oracle Virtual Desktop Client session and an application running on the local desktop.

Related Commands

- [utcrypto](#)
- [utdevadm](#)
- [utpolicy](#)

B.5.2. System Policy

This page enables you to configure group-wide policies. Some policy setting combinations are not allowed, and settings are disabled accordingly to enforce these rules.

Actions

- Set session policies for smart cards.
- Set session policies for non-smart cards.
- Enable or disable client key confirmation for client authentication.
- Enable or disable the multihead group policy.

- Enable or disable Remote Hotdesk Authentication (RHA).

Card Users Section

These policies apply to users who try to access a session with a smart card.

- **Access** - Select who can access sessions with a smart card:
 - **None** - Select to disable session access with a smart card.
 - **All Users** - Select to enable session access to all smart card users.
 - **Users with Registered Tokens** - Select to enable session access to all smart card users with a registered token. If enabled, you can also enable self-registration of tokens and if user account authentication is required.
- **Oracle Virtual Desktop Clients** - Select to enable session access on Oracle Virtual Desktop Clients with a smart card.
- **Kiosk Mode** - Select to force the user session to be the kiosk mode session (if configured) when a smart card is used.

Non-Card Users Section

These policies apply to users who try to access a session without a smart card.

- **Access** - Select who can access sessions without a smart card:
 - **None** - Select to disable session access to users without a smart card.
 - **All Users** - Select to enable session access to users without a smart card.
 - **Users with Registered Tokens** - Select to enable session access to all users without a smart card and with a registered token. If enabled, you can also enable self-registration of tokens and if user account authentication is required.
- **Oracle Virtual Desktop Clients** - Select to enable session access on Oracle Virtual Desktop Clients without a smart card.
- **Kiosk Mode** - Select to force the user session to be the kiosk mode session (if configured) without a smart card.
- **Mobile Sessions** - Select to enable Non-Smart Card Mobility (NSCM) for sessions, or hotdesking without smart cards. You can also enable the ability for users to exit from Mobile Sessions.

Client Authentication Section

- **Client Key Confirmation Required** - Select to force client key confirmation for session access if client authentication is enabled in the Security page. Once enabled, any new Sun Ray Client will be denied a regular session when first used. To allow session access, you must first inspect and confirm the submitted key as valid. You should also set the Client Authentication Security Mode to [hard](#) in the Security page, so clients that do not participate in client authentication are rejected as well.

Multihead Feature Section

- **Multihead Feature** - Select to enable the multihead group feature for the failover group. See [Section 12.2, "Multihead Groups"](#) for details.

Session Access when Hotdesking Section

- **Direct Session Access Allowed** - Select to enable direct access to a session after hotdesking when using smart cards, which is really disabling Remote Hotdesk Authentication (RHA). If you disable RHA, users won't be presented with a login screen when hotdesking. Although this reduces the time it takes for users to hotdesk, it introduces a security risk. For example, if you have a current session and someone gains access to your smart card, the user can gain access to your session without having your login information.

Related Commands

- `utreader`
- `utpolicy`

B.5.3. Kiosk Mode

This page enables you to set the kiosk session type and general properties used when kiosk mode is enabled, such as with the Windows connector or VMware View connector. This page is available only if you have configured kiosk mode as part of the initial Sun Ray Software installation or by using the `utconfig` command after post installation.

See [Chapter 10, Kiosk Mode](#) for more information.

Actions

- Configure a specific kiosk session type, including general properties and any Windows connector (`uttsc` command) arguments.

Session Type Fields

To configure a kiosk session type, fill in the following fields and click OK. Most of the fields are not required, and the system default is applied.

- **Session** - The session type to use for the Kiosk session.
- **Timeout** - Indicates the number of seconds after which a disconnected session will be terminated. If you provide no value for this setting, termination of disconnected sessions will be disabled.
- **Maximum CPU Time** - Indicates the maximum number of CPU seconds per process for kiosk sessions.
- **Maximum VM Size** - Indicates the maximum Virtual Memory size per process for kiosk sessions.
- **Maximum number of Files** - Indicates the maximum number of open files per process for kiosk sessions.
- **Maximum File Size** - Indicates the maximum file size per process for kiosk sessions.
- **Locale** - Indicates the locale to be used by the kiosk session.
- **Arguments** - Indicates a list of Windows connector (`uttsc` command) arguments that are passed to the kiosk session as it starts. This setting is specific to the kiosk session type. For more information about supported arguments, see [Chapter 10, Kiosk Mode](#).

Edit and Delete Buttons

If a kiosk session is currently configured, the Edit button is displayed, which enables you to edit the currently configured kiosk session type, and the Delete button is displayed, which enables you to delete

the currently configured kiosk session type. You can also use the Edit button to change the current kiosk session type, or you can disable kiosk mode policy to ignore the currently configured session type.

Related Commands

- `utkiosk`

B.5.4. Card Probe Order

This page enables you to set the group-wide smart card probe order, which is an ordered list of the smart card configuration files. Every time a smart card is inserted into a Sun Ray Client, the Sun Ray server tries to identify the card type using the specified probe order. Only smart cards identified by one of the configuration files specified in the probe order list are accepted. You can add or remove smart card configuration files from this list to restrict session access to specific card types.

In the absence of a group-wide probe order, the Sun Ray server uses the local probe order defined in the `/etc/opt/SUNWut/smartcard/probe_order.conf` file. If no local probe order has been set up, a default probe order is used. Changes in smart card probe order require Sun Ray services to be restarted.

See [Chapter 8, Smart Card Services](#) for more information.

Actions

- Add smart card configuration files to the group-wide smart card probe order.
- Rearrange the smart card configuration files in the smart card probe order.

Set Probe Order Button

Click this button to add, remove, and order the group-wide probe order for the smart card configuration files. The available smart card list contains the list of configuration files located in the server's `/etc/opt/SUNWut/smartcard` directory. All files end with a `.cfg` suffix, as in, `acme_card.cfg`.

Related Commands

- `utcard`

B.5.5. Data Store Password

This page enables you to change the password of the administrative user for privileged access to the Sun Ray data store. By default, the Admin GUI uses the same account to authenticate users during login. The initial password of this `admin` user is specified during the Sun Ray Software configuration.

If you change the password using the Admin GUI, the new password is applied to the Sun Ray data store as well as to the password file on the local server.



Note

In a failover group each server uses its own local password file. Thus, after changing the data store password, you must also manually update the password files on all the other servers, by using the Admin GUI or running the `utpw` command on each server.

Related Commands

- `utpw`

B.6. Log Files Tab

This tab enables you to view the following log files on the Sun Ray server that you are logged in to.

Actions

- Display Sun Ray system messages.
- Display authentication events.
- Display server administration events.
- Display mount messages.
- Display storage events.

Log File Pages

Each log file page provides a File dropdown file menu that enables you to view the current and past versions of the log file. All the log files are stored under `/var/opt/SUNWut/log`. The Size field specifies the size of the log file.

Click on each subheading to view the associated message file:

- **Messages** - Displays Sun Ray system messages (`messages.*`). The log files are archived daily.
- **Authentication** - Displays authentication events (`auth_log.*`). The log files are archived whenever the Authentication Manager is restarted.
- **Administration** - Displays operations performed during server administration (`admin_log.*`). The log files are archived daily.
- **Mount** - Displays mount messages from mass storage devices (`utmountd.log.*`). The log files are archived whenever the mounter service is restarted.
- **Storage** - Displays mass storage device events (`utstoraged.log.*`). The log files are archived whenever the storage service is restarted.

Appendix C. Third Party Licenses

This appendix includes the licenses for the third-party products included with the Sun Ray Software.

Free BSD

Copyright 1994-2009 The FreeBSD Project. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE FREEBSD PROJECT ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FREEBSD PROJECT OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. The views and conclusions contained in the software and documentation are those of the authors and should not be interpreted as representing official policies, either expressed or implied, of the FreeBSD Project.

=====

libcurl

Copyright (c) 1996 - 2010, Daniel Stenberg, <daniel@haxx.se>.
All rights reserved.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

=====

SSCEP

Copyright (c) 2003 Jarkko Turkulainen. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

```
THIS SOFTWARE IS PROVIDED BY JARKKO TURKULAINEN ``AS IS'' AND ANY EXPRESS OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND
FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL JARKKO TURKULAINEN BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
(INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING
IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH
DAMAGE.
```

```
=====
```

```
wpa_supplicant
```

```
-----
```

```
Oracle elects to license all wpa_supplicant code under the BSD license.
```

```
/*
```

```
 * WPA Supplicant
```

```
 * Copyright (c) 2003-2009, Jouni Malinen <j@w1.fi>
```

```
 *
```

```
 * This program is free software; you can redistribute it and/or modify
```

```
 * it under the terms of the GNU General Public License version 2 as
```

```
 * published by the Free Software Foundation.
```

```
 *
```

```
 * Alternatively, this software may be distributed under the terms of BSD
 * license.
```

```
 *
```

```
 * See README and COPYING for more details.
```

```
 *
```

```
 * This file implements functions for registering and unregistering
```

```
 * %wpa_supplicant interfaces. In addition, this file contains number of
```

```
 * functions for managing network connections.
```

```
 */
```

```
Redistribution and use in source and binary forms, with or without
```

```
modification, are permitted provided that the following conditions are
```

```
met:
```

```
1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
```

```
2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.
```

```
3. Neither the name(s) of the above-listed copyright holder(s) nor the
names of its contributors may be used to endorse or promote products
derived from this software without specific prior written permission.
```

```
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

```
wpa_supp/ctrl_iface_dbus.h
```

```
/*
```

```
 * WPA Supplicant / dbus-based control interface
```

```
 * Copyright (c) 2006, Dan Williams <dcbw@redhat.com> and Red Hat, Inc.
```

```
 *
```

```
 * This program is free software; you can redistribute it and/or modify
```

```
 * it under the terms of the GNU General Public License version 2 as
```



```

* published by the Free Software Foundation.
*
* Alternatively, this software may be distributed under the terms of BSD
* license.
*
* See README and COPYING for more details.
*/
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

3. Neither the name(s) of the above-listed copyright holder(s) nor the
names of its contributors may be used to endorse or promote products
derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

wpa_supp/mlme.h

/*
* WPA Supplicant - Client mode MLME
* Copyright (c) 2003-2007, Jouni Malinen <jwl.fi>
* Copyright (c) 2004, Instant802 Networks, Inc.
* Copyright (c) 2005-2006, Devicescape Software, Inc.
*
* This program is free software; you can redistribute it and/or modify
* it under the terms of the GNU General Public License version 2 as
* published by the Free Software Foundation.
*
* Alternatively, this software may be distributed under the terms of BSD
* license.
*
* See README and COPYING for more details.
*/
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

1. Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the
documentation and/or other materials provided with the distribution.

3. Neither the name(s) of the above-listed copyright holder(s) nor the
names of its contributors may be used to endorse or promote products
derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS

```

"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

common/ieee802_11_defs.h

```
/*
 * WPA Supplicant - Client mode MLME
 * Copyright (c) 2003-2007, Jouni Malinen <j@w1.fi>
 * Copyright (c) 2004, Instant802 Networks, Inc.
 * Copyright (c) 2005-2006, Devicescape Software, Inc.
 *
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License version 2 as
 * published by the Free Software Foundation.
 *
 * Alternatively, this software may be distributed under the terms of BSD
 * license.
 *
 * See README and COPYING for more details.
 */
```

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name(s) of the above-listed copyright holder(s) nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====

libXrandr

Copyright 2000, Compaq Computer Corporation,
Copyright 2002, Hewlett Packard, Inc.

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Compaq or HP not be used in advertising or publicity pertaining to distribution of the software without specific, written prior

permission. HP makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

HP DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL HP BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

=====

xrandr

Copyright 2001 Keith Packard, member of The XFree86 Project, Inc. Copyright 2002 Hewlett Packard Company, Inc.

Copyright 2006 Intel Corporation

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of the copyright holders not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. The copyright holders make no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

THE COPYRIGHT HOLDERS DISCLAIM ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

=====

OpenSSL

OpenSSL License Copyright (c) 1998-2008 The OpenSSL Project. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (<http://www.openssl.org/>)"
4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org.
5. Products derived from this software may not be called "OpenSSL" nor may "OpenSSL" appear in their names without prior written permission of the OpenSSL Project.
6. Redistributions of any form whatsoever must retain the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>)"

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND

FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com).
This product includes software written by Tim Hudson (tjh@cryptsoft.com)

Original SSLeay License

Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
All rights reserved.

This package is an SSL implementation written by Eric Young (eay@cryptsoft.com). The implementation was written so as to conform with Netscapes SSL.

This library is free for commercial and non-commercial use as long as the following conditions are aheared to. The following conditions apply to all code found in this distribution, be it the RC4, RSA, lhash, DES, etc., code; not just the SSL code. The SSL documentation included with this distribution is covered by the same copyright terms except that the holder is Tim Hudson (tjh@cryptsoft.com).

Copyright remains Eric Young's, and as such any Copyright notices in the code are not to be removed. If this package is used in a product, Eric Young should be given attribution as the author of the parts of the library used. This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement: "This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)" The word 'cryptographic' can be left out if the rouines from the library being used are not cryptographic related :-).
4. If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include an acknowledgement: "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The licence and distribution terms for any publically available version or derivative of this code cannot be changed. i.e. this code cannot simply be copied and put under another distribution licence [including the GNU Public Licence.]

=====

Assembly instructions for JPEG compression on x86:

The assembly code has a different license than the rest of the TurboJpeg package and it is

listed in simd/jsimdex.inc.

Copyright 2009 Pierre Ossman for Cendio AB
Copyright 2010 D. R. Commander

Based on
x86 SIMD extension for IJG JPEG library - version 1.02

Copyright (C) 1999-2006, MIYASAKA Masaru.

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software. Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.
2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.
3. This notice may not be removed or altered from any source distribution.

=====

The Independent JPEG Group's JPEG software:

LEGAL ISSUES

=====

In plain English:

1. We don't promise that this software works. (But if you find any bugs, please let us know!)
2. You can use this software for whatever you want. You don't have to pay us.
3. You may not pretend that you wrote this software. If you use it in a program, you must acknowledge somewhere in your documentation that you've used the IJG code.

In legalese:

The authors make NO WARRANTY or representation, either express or implied, with respect to this software, its quality, accuracy, merchantability, or fitness for a particular purpose. This software is provided "AS IS", and you, its user, assume the entire risk as to its quality and accuracy.

This software is copyright (C) 1991-2010, Thomas G. Lane, Guido Vollbeding.
All Rights Reserved except as specified below.

Permission is hereby granted to use, copy, modify, and distribute this software (or portions thereof) for any purpose, without fee, subject to these conditions:

- (1) If any part of the source code for this software is distributed, then this README file must be included, with this copyright and no-warranty notice unaltered; and any additions, deletions, or changes to the original files must be clearly indicated in accompanying documentation.
- (2) If only executable code is distributed, then the accompanying documentation must state that "this software is based in part on the work of the Independent JPEG Group".
- (3) Permission for use of this software is granted only if the user accepts full responsibility for any undesirable consequences; the authors accept NO LIABILITY for damages of any kind.

These conditions apply to any software derived from or based on the IJG code, not just to the unmodified library. If you use our work, you ought to acknowledge us.

Permission is NOT granted for the use of any IJG author's name or company name in advertising or publicity relating to this software or products derived from it. This software may be referred to only as "the Independent JPEG Group's software".

We specifically permit and encourage the use of this software as the basis of commercial products, provided that all warranty or liability claims are assumed by the product vendor.

ansi2knr.c is included in this distribution by permission of L. Peter Deutsch, sole proprietor of its copyright holder, Aladdin Enterprises of Menlo Park, CA. ansi2knr.c is NOT covered by the above copyright and conditions, but instead by the usual distribution terms of the Free Software Foundation; principally, that you must include source code if you redistribute it. (See the file ansi2knr.c for full details.) However, since ansi2knr.c is not needed as part of any program generated from the IJG code, this does not limit you more than the foregoing paragraphs do.

The Unix configuration script "configure" was produced with GNU Autoconf. It is copyright by the Free Software Foundation but is freely distributable. The same holds for its supporting scripts (config.guess, config.sub, ltmain.sh). Another support script, install-sh, is copyright by X Consortium but is also freely distributable.

The IJG distribution formerly included code to read and write GIF files. To avoid entanglement with the Unisys LZW patent, GIF reading support has been removed altogether, and the GIF writer has been simplified to produce "uncompressed GIFs". This technique does not use the LZW algorithm; the resulting GIF files are larger than usual, but are readable by all standard GIF decoders.

We are required to state that
"The Graphics Interchange Format(c) is the Copyright property of CompuServe Incorporated.
GIF(sm) is a Service Mark property of CompuServe Incorporated."

=====

JPEG Encoding:

The libjpeg-turbo toplevel license text is this:

Some of the optimizations to the Huffman encoder (jchuff.c) and decoder (jdchuff.c) were borrowed from VirtualGL, and thus any distribution of libjpeg-turbo which includes those optimizations must, as a whole, be subject to the terms of the wxWindows Library Licence, Version 3.1. A copy of this license can be found in this directory under LICENSE.txt. The wxWindows Library License is based on the LGPL but includes provisions which allow the Library to be statically linked into proprietary libraries and applications without requiring the resulting binaries to be distributed under the terms of the LGPL.

The rest of the source code, apart from the Huffman codec optimizations, falls under a less restrictive, BSD-style license (see README.) You can choose to distribute libjpeg-turbo, as a whole, under this BSD-style license by simply replacing the optimized jchuff.c and jdchuff.c with their unoptimized counterparts from the libjpeg v6b source.

We only use the code that falls under the less restrictive BSD style license, the text for that is:

LEGAL ISSUES

=====

In plain English:

1. We don't promise that this software works. (But if you find any bugs, please let us know!)
2. You can use this software for whatever you want. You don't have to pay us.
3. You may not pretend that you wrote this software. If you use it in a program, you must acknowledge somewhere in your documentation that you've used the IJG code.

In legalese:

The authors make NO WARRANTY or representation, either express or implied, with respect to this software, its quality, accuracy, merchantability, or fitness for a particular purpose. This software is provided "AS IS", and you, its user, assume the entire risk as to its quality and accuracy.

This software is copyright (C) 1991-1998, Thomas G. Lane.
All Rights Reserved except as specified below.

Permission is hereby granted to use, copy, modify, and distribute this software (or portions thereof) for any purpose, without fee, subject to these conditions:

(1) If any part of the source code for this software is distributed, then this README file must be included, with this copyright and no-warranty notice unaltered; and any additions, deletions, or changes to the original files must be clearly indicated in accompanying documentation.

(2) If only executable code is distributed, then the accompanying documentation must state that "this software is based in part on the work of the Independent JPEG Group".

(3) Permission for use of this software is granted only if the user accepts full responsibility for any undesirable consequences; the authors accept NO LIABILITY for damages of any kind.

These conditions apply to any software derived from or based on the IJG code, not just to the unmodified library. If you use our work, you ought to acknowledge us.

Permission is NOT granted for the use of any IJG author's name or company name in advertising or publicity relating to this software or products derived from it. This software may be referred to only as "the Independent JPEG Group's software".

We specifically permit and encourage the use of this software as the basis of commercial products, provided that all warranty or liability claims are assumed by the product vendor. ansi2knr.c is included in this distribution by permission of L. Peter Deutsch, sole proprietor of its copyright holder, Aladdin Enterprises of Menlo Park, CA.

ansi2knr.c is NOT covered by the above copyright and conditions, but instead by the usual distribution terms of the Free Software Foundation; principally, that you must include source code if you redistribute it. (See the file ansi2knr.c for full details.) However, since ansi2knr.c is not needed as part of any program generated from the IJG code, this does not limit you more than the foregoing paragraphs do.

The Unix configuration script "configure" was produced with GNU Autoconf. It is copyright by the Free Software Foundation but is freely distributable. The same holds for its supporting scripts (config.guess, config.sub, ltconfig, ltmain.sh). Another support script, install-sh, is copyright by M.I.T. but is also freely distributable.

It appears that the arithmetic coding option of the JPEG spec is covered by patents owned by IBM, AT&T, and Mitsubishi. Hence arithmetic coding cannot legally be used without obtaining one or more licenses. For this reason, support for arithmetic coding has been removed from the free JPEG software. (Since arithmetic coding provides only a marginal gain over the unpatented Huffman mode, it is unlikely that very many implementations will support it.) So far as we are aware, there are no patent restrictions on the remaining code.

The IJG distribution formerly included code to read and write GIF files. To avoid entanglement with the Unisys LZW patent, GIF reading support has been removed altogether, and the GIF writer has been simplified to produce "uncompressed GIFs". This technique does not use the LZW algorithm; the resulting GIF files are larger than usual, but are readable by all standard GIF decoders.

We are required to state that

"The Graphics Interchange Format(c) is the Copyright property of CompuServe Incorporated. GIF(sm) is a Service Mark property of CompuServe Incorporated."

=====

Cryptolib:

* This is version 1.2 of CryptoLib

*

* The authors of this software are Jack Lacy, Don Mitchell and Matt Blaze

* Copyright (c) 1991, 1992, 1993, 1994, 1995 by AT&T.

* Permission to use, copy, and modify this software without fee

* is hereby granted, provided that this entire notice is included in

* all copies of any software which is or includes a copy or

```

* modification of this software and in all copies of the supporting
* documentation for such software.
*
* NOTE:
* Some of the algorithms in cryptolib may be covered by patents.
* It is the responsibility of the user to ensure that any required
* licenses are obtained.
*
*
* SOME PARTS OF CRYPTOLIB MAY BE RESTRICTED UNDER UNITED STATES EXPORT
* REGULATIONS.
*
*
* THIS SOFTWARE IS BEING PROVIDED "AS IS", WITHOUT ANY EXPRESS OR IMPLIED
* WARRANTY. IN PARTICULAR, NEITHER THE AUTHORS NOR AT&T MAKE ANY
* REPRESENTATION OR WARRANTY OF ANY KIND CONCERNING THE MERCHANTABILITY
* OF THIS SOFTWARE OR ITS FITNESS FOR ANY PARTICULAR PURPOSE.
?

1. Rijmen.x: Vincent Rijmen "This code is hereby placed in the public domain"
2. MD5.c: RSA Data Security Inc., 1991
3. des_xxx and podd.c: Eric Young 1995-98 (various similar but not identical forms and years
depending on the source module) Copies of the notices in the code follow:

=====

1. Vincent Rijmen "This code is hereby placed in the public domain"
-----

/*
 * @author Vincent Rijmen
 * @author Antoon Bosselaers
 * @author Paulo Barreto
 *
 * This code is hereby placed in the public domain.
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHORS 'AS IS' AND ANY EXPRESS
 * OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
 * WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS OR CONTRIBUTORS BE
 * LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
 * CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
 * SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
 * BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
 * WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE
 * OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE,
 * EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */

=====

2. RSA Data Security Inc., 1991
-----

/* MD5C.C - RSA Data Security, Inc., MD5 message-digest algorithm */
/* Copyright (C) 1991, RSA Data Security, Inc. All rights reserved. License to copy and use
this software is granted provided that it is identified as the "RSA Data Security, Inc. MD5
Message-Digest Algorithm" in all material mentioning or referencing this software or this
function. License is also granted to make and use derivative works provided that such works
are identified as "derived from the RSA Data Security, Inc. MD5 Message-Digest Algorithm" in
all material mentioning or referencing the derived work. RSA Data Security, Inc. makes no
representations concerning either the merchantability of this software or the suitability of
this software for any particular purpose. It is provided "as is" without express or implied
warranty of any kind. These notices must be retained in any copies of any part of this
documentation and/or software. */

```

=====

3. Eric Young 1995-98 (various forms and years depending on the source module)

Eric Young 1

des.h

```
/* $NetBSD: des.h,v 1.5 2001/09/09 11:01:02 tls Exp $ */
/* $KAME: des.h,v 1.7 2000/09/18 20:59:21 itojun Exp $ */
/* lib/des/des.h */
/* Copyright (C) 1995-1996 Eric Young (eay@mincom.oz.au)
 * All rights reserved.
 *
 * This file is part of an SSL implementation written
 * by Eric Young (eay@mincom.oz.au).
 * The implementation was written so as to conform with Netscapes SSL
 * specification. This library and applications are
 * FREE FOR COMMERCIAL AND NON-COMMERCIAL USE
 * as long as the following conditions are aheared to.
 *
 * Copyright remains Eric Young's, and as such any Copyright notices in
 * the code are not to be removed. If this code is used in a product,
 * Eric Young should be given attribution as the author of the parts used.
 * This can be in the form of a textual message at program startup or
 * in documentation (online or textual) provided with the package.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 * 3. All advertising materials mentioning features or use of this software
 * must display the following acknowledgement:
 * This product includes software developed by Eric Young (eay@mincom.oz.au)
 *
 * THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 *
 * The licence and distribution terms for any publically available version or
 * derivative of this code cannot be changed. i.e. this code cannot simply be
 * copied and put under another distribution licence
 * [including the GNU Public Licence.]
 */
```

Eric Young 2

des_cbc.c

```
/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
 * All rights reserved.
 *
 * This package is an SSL implementation written
 * by Eric Young (eay@cryptsoft.com).
 * The implementation was written so as to conform with Netscapes SSL.
```

```

*
* This library is free for commercial and non-commercial use as long as
* the following conditions are aheared to. The following conditions
* apply to all code found in this distribution, be it the RC4, RSA,
* lhash, DES, etc., code; not just the SSL code. The SSL documentation
* included with this distribution is covered by the same copyright terms
* except that the holder is Tim Hudson (tjh@cryptsoft.com).
*
* Copyright remains Eric Young's, and as such any Copyright notices in
* the code are not to be removed.
* If this package is used in a product, Eric Young should be given attribution
* as the author of the parts of the library used.
* This can be in the form of a textual message at program startup or
* in documentation (online or textual) provided with the package
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. All advertising materials mentioning features or use of this software
* must display the following acknowledgement:
* "This product includes cryptographic software written by
* Eric Young (eay@cryptsoft.com)"
* The word 'cryptographic' can be left out if the rouines from the library
* being used are not cryptographic related :-).
* 4. If you include any Windows specific code (or a derivative thereof) from
* the apps directory (application code) you must include an acknowledgement:
* "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
*
* THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*
* The licence and distribution terms for any publically available version or
* derivative of this code cannot be changed. i.e. this code cannot simply be
* copied and put under another distribution licence
* [including the GNU Public Licence.]
*/

Eric Young 3
-----

/* $NetBSD: des_ecb.c,v 1.7 2002/11/02 07:19:51 perry Exp $ */
/* $KAME: des_ecb.c,v 1.5 2000/11/06 13:58:08 itojun Exp $ */
/* crypto/des/ecb_enc.c */
/* Copyright (C) 1995-1998 Eric Young (eay@mincom.oz.au)
 * All rights reserved.
 *
 * This file is part of an SSL implementation written
 * by Eric Young (eay@mincom.oz.au).
 * The implementation was written so as to conform with Netscapes SSL
 * specification. This library and applications are
 * FREE FOR COMMERCIAL AND NON-COMMERCIAL USE
 * as long as the following conditions are aheared to.
 *

```

```

* Copyright remains Eric Young's, and as such any Copyright notices in
* the code are not to be removed. If this code is used in a product,
* Eric Young should be given attribution as the author of the parts used.
* This can be in the form of a textual message at program startup or
* in documentation (online or textual) provided with the package.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. All advertising materials mentioning features or use of this software
* must display the following acknowledgement:
* This product includes software developed by Eric Young (eay@mincom.oz.au)
*
* THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*
* The licence and distribution terms for any publically available version or
* derivative of this code cannot be changed. i.e. this code cannot simply be
* copied and put under another distribution licence
* [including the GNU Public Licence.]
*/

Eric Young 4
-----

/* crypto/des/des_enc.c */
/* Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com)
 * All rights reserved.
 *
 * This package is an SSL implementation written
 * by Eric Young (eay@cryptsoft.com).
 * The implementation was written so as to conform with Netscapes SSL.
 *
 * This library is free for commercial and non-commercial use as long as
 * the following conditions are aheared to. The following conditions
 * apply to all code found in this distribution, be it the RC4, RSA,
 * lhash, DES, etc., code; not just the SSL code. The SSL documentation
 * included with this distribution is covered by the same copyright terms
 * except that the holder is Tim Hudson (tjh@cryptsoft.com).
 *
 * Copyright remains Eric Young's, and as such any Copyright notices in
 * the code are not to be removed.
 * If this package is used in a product, Eric Young should be given attribution
 * as the author of the parts of the library used.
 * This can be in the form of a textual message at program startup or
 * in documentation (online or textual) provided with the package.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright

```

```

* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. All advertising materials mentioning features or use of this software
* must display the following acknowledgement:
* "This product includes cryptographic software written by
* Eric Young (eay@cryptsoft.com)"
* The word 'cryptographic' can be left out if the routines from the library
* being used are not cryptographic related :-).
* 4. If you include any Windows specific code (or a derivative thereof) from
* the apps directory (application code) you must include an acknowledgement:
* "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"
*
* THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*
* The licence and distribution terms for any publically available version or
* derivative of this code cannot be changed. i.e. this code cannot simply be
* copied and put under another distribution licence
* [including the GNU Public Licence.]
*/

Eric Young 5
-----

/* $NetBSD: des_locl.h,v 1.4 2001/09/09 11:01:02 tls Exp $ */
/* $KAME: des_locl.h,v 1.6 2000/11/06 13:58:09 itojun Exp $ */
/* crypto/des/des_locl.h */
/* Copyright (C) 1995-1997 Eric Young (eay@mincom.oz.au)
 * All rights reserved.
 *
 * This file is part of an SSL implementation written
 * by Eric Young (eay@mincom.oz.au).
 * The implementation was written so as to conform with Netscapes SSL
 * specification. This library and applications are
 * FREE FOR COMMERCIAL AND NON-COMMERCIAL USE
 * as long as the following conditions are adhered to.
 *
 * Copyright remains Eric Young's, and as such any Copyright notices in
 * the code are not to be removed. If this code is used in a product,
 * Eric Young should be given attribution as the author of the parts used.
 * This can be in the form of a textual message at program startup or
 * in documentation (online or textual) provided with the package.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 * 3. All advertising materials mentioning features or use of this software
 * must display the following acknowledgement:
 * This product includes software developed by Eric Young (eay@mincom.oz.au)
 *
 * THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE

```

```

* ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*
* The licence and distribution terms for any publically available version or
* derivative of this code cannot be changed. i.e. this code cannot simply be
* copied and put under another distribution licence
* [including the GNU Public Licence.]
*/

Eric Young 6
-----

/* $NetBSD: des_setkey.c,v 1.8 2002/11/07 07:04:13 thorpej Exp $ */
/* $KAME: des_setkey.c,v 1.6 2001/07/03 14:27:53 itojun Exp $ */
/* crypto/des/set_key.c */
/* Copyright (C) 1995-1996 Eric Young (eay@mincom.oz.au)
 * All rights reserved.
 *
 * This file is part of an SSL implementation written
 * by Eric Young (eay@mincom.oz.au).
 * The implementation was written so as to conform with Netscapes SSL
 * specification. This library and applications are
 * FREE FOR COMMERCIAL AND NON-COMMERCIAL USE
 * as long as the following conditions are aheared to.
 *
 * Copyright remains Eric Young's, and as such any Copyright notices in
 * the code are not to be removed. If this code is used in a product,
 * Eric Young should be given attribution as the author of the parts used.
 * This can be in the form of a textual message at program startup or
 * in documentation (online or textual) provided with the package.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 * 3. All advertising materials mentioning features or use of this software
 * must display the following acknowledgement:
 * This product includes software developed by Eric Young (eay@mincom.oz.au)
 *
 * THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 *
 * The licence and distribution terms for any publically available version or
 * derivative of this code cannot be changed. i.e. this code cannot simply be
 * copied and put under another distribution licence
 * [including the GNU Public Licence.]
*/

```

Eric Young 7

```
-----
/* $NetBSD: podd.h,v 1.1 2000/06/14 19:45:36 thorpej Exp $ */
/* $KAME: podd.h,v 1.3 2000/03/27 04:36:34 sumikawa Exp $ */
/* crypto/des/podd.h */
/* Copyright (C) 1995-1996 Eric Young (eay@mincom.oz.au)
 * All rights reserved.
 *
 * This file is part of an SSL implementation written
 * by Eric Young (eay@mincom.oz.au).
 * The implementation was written so as to conform with Netscapes SSL
 * specification. This library and applications are
 * FREE FOR COMMERCIAL AND NON-COMMERCIAL USE
 * as long as the following conditions are aheared to.
 *
 * Copyright remains Eric Young's, and as such any Copyright notices in
 * the code are not to be removed. If this code is used in a product,
 * Eric Young should be given attribution as the author of the parts used.
 * This can be in the form of a textual message at program startup or
 * in documentation (online or textual) provided with the package.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the copyright
 * notice, this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 * notice, this list of conditions and the following disclaimer in the
 * documentation and/or other materials provided with the distribution.
 * 3. All advertising materials mentioning features or use of this software
 * must display the following acknowledgement:
 * This product includes software developed by Eric Young (eay@mincom.oz.au)
 *
 * THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 *
 * The licence and distribution terms for any publically available version or
 * derivative of this code cannot be changed. i.e. this code cannot simply be
 * copied and put under another distribution licence
 * [including the GNU Public Licence.]
 */
=====

libusb:
-----

Oracle elects to license libusb under the BSD license.

Copyright (c) 2000-2003 Johannes Erdfelt
This file (and only this file) may alternatively be licensed under the BSD license as well,
read LICENSE for details.

Redistribution and use in source and binary forms, with or without modification, are
permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of
conditions and the following disclaimer.
```

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====

XWindow:

Copyright 1987, 1998? The Open Group

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation.

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.? IN NO EVENT SHALL THE OPEN GROUP BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of The Open Group shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from The Open Group.

Copyright 1987 by Digital Equipment Corporation, Maynard, Massachusetts.

All Rights Reserved

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of Digital not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

DIGITAL DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL DIGITAL BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

*****/
/* The panoramix components contained the following notice */
/*****

Copyright (c) 1991, 1997 Digital Equipment Corporation, Maynard, Massachusetts.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge,

publish, distribute, sublicense, and/or sell copies of the Software.

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.? IN NO EVENT SHALL DIGITAL EQUIPMENT CORPORATION BE LIABLE FOR ANY CLAIM, DAMAGES, INCLUDING, BUT NOT LIMITED TO CONSEQUENTIAL OR INCIDENTAL DAMAGES, OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of Digital Equipment Corporation shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization from Digital Equipment Corporation. Newer files in the distribution have licenses similar to this one, except the author/organization may be different:

```
/*
?* Copyright ? 2000 Compaq Computer Corporation
?* Copyright ? 2002 Hewlett-Packard Company
?* Copyright ? 2006 Intel Corporation
?*
?* Permission to use, copy, modify, distribute, and sell this software and its
?* documentation for any purpose is hereby granted without fee, provided that
?* the above copyright notice appear in all copies and that both that copyright
?* notice and this permission notice appear in supporting documentation, and
?* that the name of the copyright holders not be used in advertising or
?* publicity pertaining to distribution of the software without specific,
?* written prior permission.? The copyright holders make no representations
?* about the suitability of this software for any purpose.? It is provided "as
?* is" without express or implied warranty.
?*
?* THE COPYRIGHT HOLDERS DISCLAIM ALL WARRANTIES WITH REGARD TO THIS SOFTWARE,
?* INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO
?* EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY SPECIAL, INDIRECT OR
?* CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE,
?* DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER
?* TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE
?* OF THIS SOFTWARE.
?*
?* Author:
?* Jim Gettys, Hewlett-Packard Company, Inc.
?*????????? Keith Packard, Intel Corporation ?
*/
```

=====

TurboJPEG:

Copyright (C) 1998-2005 Julian Smart, Robert Roebling et al

Everyone is permitted to copy and distribute verbatim copies of this licence document, but changing it is not allowed.

WXWINDOWS LIBRARY LICENCE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Library General Public Licence as published by the Free Software Foundation; either version 2 of the Licence, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Library General Public Licence for more details.

You should have received a copy of the GNU Library General Public Licence along with this software, usually in a file named COPYING.LIB. If not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA.

EXCEPTION NOTICE

1. As a special exception, the copyright holders of this library give permission for additional uses of the text contained in this release of the library as licenced under the wxWindows Library Licence, applying either version 3.1 of the Licence, or (at your option) any later version of the Licence as published by the copyright holders of version 3.1 of the Licence document.

2. The exception is that you may use, copy, link, modify and distribute under your own terms, binary object code versions of works based on the Library.

3. If you copy code from files distributed under the terms of the GNU General Public Licence or the GNU Library General Public Licence into a copy of this library, as this licence permits, the exception does not apply to the code that you add in this way. To avoid misleading anyone as to the status of such modified files, you must delete this exception notice from such code and/or adjust the licensing conditions notice accordingly.

4. If you write modifications of your own for this library, it is your choice whether to permit this exception to apply to your modifications. If you do not wish that, you must delete the exception notice from such code and/or adjust the licensing conditions notice accordingly.

=====

IPSEC ipsec_tools:

```
* Copyright (C) 1995, 1996, 1997, and 1998 WIDE Project.
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. Neither the name of the project nor the names of its contributors
* may be used to endorse or promote products derived from this software
* without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE PROJECT AND CONTRIBUTORS ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE PROJECT OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*/
```

Some files have:

```
/*
* Copyright (C) 2004 Emmanuel Dreyfus
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
```

```

* are met:
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. Neither the name of the project nor the names of its contributors
* may be used to endorse or promote products derived from this software
* without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE PROJECT AND CONTRIBUTORS ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE PROJECT OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*/

and some files have:

/*
* Copyright (C) 2004 SuSE Linux AG, Nuernberg, Germany.
* Contributed by: Michal Ludvig , SUSE Labs
* All rights reserved.
*
* Redistribution and use in source and binary forms, with or without
* modification, are permitted provided that the following conditions
* are met:
* 1. Redistributions of source code must retain the above copyright
* notice, this list of conditions and the following disclaimer.
* 2. Redistributions in binary form must reproduce the above copyright
* notice, this list of conditions and the following disclaimer in the
* documentation and/or other materials provided with the distribution.
* 3. Neither the name of the project nor the names of its contributors
* may be used to endorse or promote products derived from this software
* without specific prior written permission.
*
* THIS SOFTWARE IS PROVIDED BY THE PROJECT AND CONTRIBUTORS ``AS IS'' AND
* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
* IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
* ARE DISCLAIMED. IN NO EVENT SHALL THE PROJECT OR CONTRIBUTORS BE LIABLE
* FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
* DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
* OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
* LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
* OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
*/

=====

PC/SC-lite:
-----
Copyright (c) 1999-2003 David Corcoran
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are
permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of
conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of

```

conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission. Changes to this license can be made only by the copyright author with explicit written consent.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

=====

Glossary

A

alias token

An alias token enables a card owner to access the same Sun Ray session with more than one physical token. This setup can be useful, for example, when a user needs a duplicate smart card.

ALP

The Sun Appliance Link Protocol, a suite of network protocols that enable communication between Sun Ray servers and Sun Ray Clients.

AMGH

Automatic Multigroup Hotdesking. See *regional hotdesking*.

authentication policy

The Authentication Manager, using the selected authentication modules, decides what tokens are valid and which users, as token owners, have access to the system and sessions.

authentication token

Although all tokens are used by the Authentication Manager to grant or deny access to Sun Ray sessions, this term usually refers to a user's smart card token. See *token*.

B

backplane bandwidth

Sometimes also referred to as switch fabric. A switch's backplane is the pipe through which data flows from an input port to an output port. Backplane bandwidth usually refers to the aggregate bandwidth available amongst all ports within a switch.

barrier mechanism

To prevent clients from downloading firmware that is older than the firmware they already have, the administrator can set a barrier mechanism. The barrier mechanism symbol `BarrierLevel` is defined by default in the DHCP table of Sun Ray servers.

bpp

Bits per pixel.

C

CAM

Controlled Access Mode, was renamed to [kiosk mode](#).

card reader

See [token reader](#).

client

See [desktop client](#).

client ID

The unique identifier for a client. For Sun Ray Clients, it is the client's MAC address. For Oracle Virtual Desktop Clients, it is an MD5 hash of the client key. Client ID is also referred to as CID, terminal CID, client identifier, or desktop ID.

client key

An automatically generated public-private key pair that represents a Sun Ray Client or an Oracle Virtual Desktop Client. A client key is used to authenticate the device when it connects to a server.

client-server

A common way to describe network services and the user processes (programs) of those services.

codec

A device or program capable of encoding or decoding a digital data stream or signal.

cold restart

See [restart](#).

Configuration GUI

A tool to modify a Sun Ray Client's local configuration for initialization and booting.

D

desktop client

In Sun Ray Software documentation, used to refer to both a [Sun Ray Client](#) or [Oracle Virtual Desktop Client](#). The abbreviated term [client](#) is also used.

DHCP

Dynamic Host Configuration Protocol, which is a means of distributing IP addresses and initial parameters to the clients.

display

One or more screens from a single Sun Ray session.

DTU

See [Sun Ray Client](#).

dynamic session resizing

A feature that allows the remote desktop to be resized automatically to fit the optimized size of your local desktop client session. When you hotdesk to a session from a different device, or use a client device like a tablet, which can be rotated, the new screen configuration is detected and the session screen dimensions are adapted accordingly.

F

failover

The process of transferring processes from a failed Sun Ray server to a functional Sun Ray server.

failover group

Two or more Sun Ray servers configured to provide continuity of service in the event of a network or system failure. Sometimes abbreviated as FOG or HA (for *high availability*). The term *high availability* refers to the benefit of this type of configuration; the term failover group refers to the functionality.

firmware

A small piece of software residing on Sun Ray Clients that handles power-on self test (POST), client initialization, authentication, and low-level input and output. See [Sun Ray Operating Software](#).

firmware barrier

See [barrier mechanism](#).

FOG

See [failover group](#).

fps

Frames per second.

frame buffer

Video output device that drives the video display. See [virtual frame buffer](#).

G

group-wide

Across a failover group.

H

HA

High availability. See [failover group](#).

head

A Sun Ray Client, with one or two monitors, used in a multihead group.

high availability

See [failover group](#).

hot key

A predefined key that causes an activity to occur. For example, a hot key is used to display the Settings screen on the Sun Ray Client.

hot-pluggable

A property of a hardware component that can be inserted into or removed from a system that is powered on. USB devices connected to Sun Ray Clients are hot-pluggable.

hotdesking

The ability for a user to remove a smart card, insert it into any other client within a failover group, and have the user's session "follow" the user, thus allowing the user to have instantaneous access to the user's windowing environment and current applications from multiple clients.

I

idle session

A session that is running on a Sun Ray server but to which no user (identified by a smart card token or a pseudo-token) is logged in.

K

keyboard country code

A number representing a specific USB keyboard map that can be set in the Sun Ray client firmware to provide better Non-US keyboard support. This code is used if the keyboard returns a country code of 0.

kiosk mode

A facility to deliver an almost unlimited variety of desktops or applications to users, even though the actual desktop or application may be running elsewhere. Kiosk mode bypasses the normal authentication methods of the platform and runs anything that the administrator defines. Kiosk sessions are configured through a Kiosk session type.

kiosk session

A user session running in kiosk mode. Also called kiosk mode session.

kiosk session type

A set of scripts and configuration files, which are described by a kiosk session descriptor file. A kiosk session type defines the kind of user session that will run in kiosk mode. A session type is sometimes referred to as a session configuration.

M

mobile token

If mobile sessions are enabled, a user can log into an existing session from different locations without a smart card, in which case the user name is associated with the session. This type of [pseudo-token](#) is called a mobile token.

mobility

For the purposes of the Sun Ray Software, the property of a session that allows it to follow a user from one client to another within a server group. On the Sun Ray system, mobility requires the use of a smart card or other identifying mechanism.

monitor

The physical monitor connected to a client.

MTU

Maximum Transmission Unit, used to specify the number of bytes in the largest packet a network can transmit.

multi-monitor

A type of multiple monitor configuration that supports multiple monitors connected to the dual video connectors on a Sun Ray 2FS or Sun Ray 3 Plus Client. By using RandR 1.2, the multiple monitors are managed as one screen.

multicasting

The process of enabling communication between Sun Ray servers over their Sun Ray network interfaces in a failover environment.

multihead group

A type of multiple monitor configuration that enables you to merge and control multiple Sun Ray Clients, referred to in this context as heads, and their screens using a single keyboard and mouse attached to a primary client.

N

network latency

The time delay associated with moving information through a network. Interactive applications such as voice, video displays and multimedia applications are sensitive to these delays.

non-smart card mobility

NSCM. A mobile session on a Sun Ray Client that does not rely on a smart card. NSCM requires a policy that allows *pseudo-token*.

NSCM

See [non-smart card mobility](#).

O

offline

A specific mode for a server in a failover group, which means the server does not participate in load balancing any more (the load balancing algorithm does not select this server for new sessions). New sessions can still be manually created on it.

Oracle Virtual Desktop Client

A software application that runs on a common client operating system and provides the ability to connect to a desktop session running on a Sun Ray server. It is a software version of a Sun Ray Client. The desktop running the application is also referred to as an Oracle Virtual Desktop Client in this document.

OSD

On-screen display. The Sun Ray Client uses OSD icons to alert the user about potential start-up or connectivity problems.

output

A single instance of a physical monitor. Each output has a physical video connector.

P

PAM

Pluggable Authentication Module. A set of dynamically loadable objects that gives system administrators the flexibility of choosing among available user authentication services.

PAM session

A single PAM handle and runtime state associated with all PAM items, data, and the like.

policy

See [authentication policy](#).

Pop-up GUI

See [configuration GUI](#).

power cycling

Using the power cord to restart a client.

private network

A network configuration where Sun Ray Clients are directly connected to the Sun Ray server, that is, the server has a network interface connected to the subnet and the server is devoted entirely to carrying Sun Ray traffic. Also known as directly-connected dedicated interconnect or private interconnect.

pseudo-session

A Sun Ray session associated with a [pseudo-token](#) rather than a smart card token.

pseudo-token

A user accessing a Sun Ray session without a smart card is identified by the client's built-in type and MAC address, known as a pseudo-token. See [token](#).

R

RDP

Microsoft Remote Desktop Protocol.

RDS

Remote Desktop Services. Formally known as Terminal Services. See [Windows Terminal Services](#).

regional hotdesking

Enables users to access their sessions across wider domains and greater physical distances. You can enable this feature by defining how user sessions are mapped to an expanded list of servers in multiple failover groups. Originally known as Automatic Multigroup Hotdesking (AMGH).

restart

Sun Ray services can be restarted either from the `utstart` command or with the Warm Restart or Cold Restart buttons through the Admin GUI. A cold restart terminates all Sun Ray sessions; a warm restart does not.

RHA

Remote Hotdesk Authentication, a security enhancement that requires Sun Ray Software authentication before users can reconnect to an existing session. RHA does not apply to kiosk sessions, which are designed for anonymous access without authentication. RHA policy can be administered either through the Admin GUI or with the `utpolicy` command.

S

screen

A monitor or group of monitors that show a single desktop to a user. A screen can be provided by a single monitor or by multiple monitors on a Sun Ray Client with dual video connectors, such as the Sun Ray 3 Plus Client. A multihead group can also show a single desktop when using Xinerama.

screen flipping

The ability to pan to individual screens that were originally created by a multihead group on a client with a single head.

service

For the purposes of Sun Ray Software, any application that can directly connect to the Sun Ray Client. It can include audio, video, X servers, access to other machines, and device control of the client.

session

A group of services or applications associated with an authentication token. Desktop sessions reside on a Sun Ray server and can be directed to any Sun Ray Client or Oracle Virtual Desktop Client. From a user's perspective, a desktop session is usually an instance of an OS desktop. A session may be associated with a token embedded on a smart card. Also known as desktop session or Sun Ray session. See [token](#).

session mobility

See [mobility](#).

smart card

Generically, a plastic card containing a microprocessor capable of making calculations. Smart cards that can be used to initiate or connect to Sun Ray sessions contain identifiers such as the card type and ID. Smart card tokens may also be registered in the Sun Ray data store, either by the Sun Ray administrator or, if the administrator chooses, by the user.

smart card-based authentication

Using a smart card to authenticate a card holder based on credentials supplied by the card and authentication information from the card holder, such as a PIN or biometric data.

smart card-based session mobility

Using a smart card to provide a unique token ID and token type that enables Sun Ray Software to locate the card holder's session. In some cases, card holders might be required to authenticate themselves using smart card-based authentication.

smart token

An authentication token contained on a smart card. See [token](#).

Sun Ray Client

A hardware client that obtains a desktop session from a Sun Ray server. The software client counterpart is called an Oracle Virtual Desktop Client. Previously referred to as Sun Ray thin clients, Sun Ray virtual display terminals, and Sun Ray DTUs (Desktop Terminal Units).

Sun Ray Operating Software

The name of the Sun Ray Client firmware. See [firmware](#).

Sun Ray system

The Sun Ray system consists of Sun Ray Clients, servers, server software, and the physical networks that connect them.

T

thin client

Thin clients remotely access some resources of a computer server such as compute power and large memory capacity. The Sun Ray Clients rely on the server for all computing power and storage.

token

The Sun Ray system requires each user to present a token that the Authentication Manager uses to allow or deny access to the system and to sessions. A token consists of a type and an ID. If the user uses a smart card, the smart card's type and ID are used as the token. If the user is not using a smart card, the client's built-in type and ID (the unit's Ethernet, or MAC, address) are used instead as a [pseudo-token](#). If mobile sessions are enabled, a user can log into an existing session from different locations without a smart card, in which case the user name is associated with the session. A pseudo-token used for mobile sessions is called a [mobile token](#). *Alias tokens* can also be created to enable users to access the same session with more than one physical token.

token reader

A Sun Ray Client that is dedicated to reading smart cards and returning their identifiers, which can be associated with card owners (that is, with users).

trusted-server

Servers in the same failover group that "trust" one another through a common group signature.

U

USB redirection

A Sun Ray Software feature that enables users to access USB devices connected to a Sun Ray Client from their Windows sessions, provided that the appropriate device drivers are installed on the Windows system.

user session

A session that is running on a Sun Ray server and to which a user, identified by a smart card token or a pseudo token, is logged in.

V

video acceleration

A feature provided in the Windows connector to improve video playback performance, which consists of the multimedia redirection and Adobe Flash acceleration components.

virtual desktop

A virtual machine containing a desktop instance that is executed and managed within the virtual desktop infrastructure, usually a Windows desktop accessed through *RDP*.

virtual frame buffer

A region of memory on the Sun Ray server that contains the current state of a user's display.

VMware View connector

Enables Sun Ray Client users to connect to Windows virtual machines through the VMware View Manager.

W

warm restart

See [restart](#).

Windows connector

A Microsoft Remote Desktop Protocol (RDP) client that enables Sun Ray users to access applications running on remote Microsoft Windows systems.

Windows system

A generic term used throughout the Sun Ray Software documentation to indicate a remote desktop server running the Windows OS that can be accessed remotely from a Sun Ray Client using the Windows connector. See [Windows Terminal Services](#) for the different ways a remote desktop is provided based on the Windows OS.

Windows Terminal Services

A Microsoft Windows component that makes Windows applications and desktops accessible to remote users and clients. Depending on the Windows release, this feature may be called Terminal Services, Remote Desktop Services, or Remote Desktop Connection.

X

X server

A process that controls a bitmap display device in an X Window System. It performs operations on request from client applications. Sun Ray Software contains two X servers: Xsun, which was the default Xserver in previous versions of Sun Ray Software on Oracle Solaris 10, and Xnewt, which is the default Xserver for Sun Ray Software on all platforms. Xnewt enables the latest multimedia capabilities.

Xinerama

An extension to the X Window System that enables the use of two or more monitors as one large virtual display. Xinerama mode allows the display of a single desktop across multiple monitors.

Xnewt

The default X server for Sun Ray Software on Oracle Solaris.

xrandr

The X Resize, Rotate and Reflect extension to the X Window System, which enables clients to resize, rotate, and change screen resolution settings dynamically. For Sun Ray Software, this extension is especially useful when a user hotdesks to Sun Ray Clients that use monitors of different sizes or resolutions than the one where a given session began.