



Diskless Setup for the Solaris OS for x86 Platforms

Vijay S. Upreti

October 2007

Sun Microsystems, Inc.

Copyright © 2007 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved.

U.S. Government Rights - Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements. Use is subject to license terms. This distribution may include materials developed by third parties.

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

All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the United States and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

Sun, Sun Microsystems, the Sun logo, Java, Solaris, Sun Blade, and Sun Fire are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

This product is covered and controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical biological weapons or nuclear maritime end uses or end users, whether direct or indirect, are strictly prohibited. Export or reexport to countries subject to U.S. embargo or to entities identified on U.S. export exclusion lists, including, but not limited to, the denied persons and specially designated nationals lists is strictly prohibited.

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

Table of Contents

Introduction.....	4
Setting Up the Environment.....	4
Adding OS Services for Diskless Client Support.....	6
Adding a Diskless Client Boot Environment.....	7
Distributing the TFTP/DHCP and NFS/Root Environment.....	9
Setting Up the DHCP Server.....	10
Resolving Issues.....	13
Running Diagnosis Commands.....	14
References.....	14
For More Information.....	15
About the Author	15
Acknowledgements.....	15
Appendix	16
A. Example dhcp_config Script.....	16
B. config_tftp File.....	17
C. Finding and Adding Missing Packages.....	20
D. Enabling the TFTP Service.....	20

Introduction

This document describes the steps for setting up a diskless installation for the Solaris™ Operating System in an x86 environment. Also described are the steps for distributing the TFTP/DHCP and NFS/root environment on separate servers. Additionally, this article describes problems that could occur during setup and how to resolve them.

Starting with the Solaris 10 1/06 release, use the procedures in this document to prepare for adding a diskless client. The procedures include general information for x86 based systems.

Parts of this document are based on Sun production documentation; see the “References” section for more information.

Setting Up the Environment

1. Configure your environment variables as follows:

```
JAVA_HOME=/usr/j2se
export JAVA_HOME
LD_LIBRARY_PATH=/usr/sadm/lib/wbem:/usr/snadm/lib
export LD_LIBRARY_PATH
```

2. Ensure that the system that is intended to be the OS service is running a supported release. Also, verify that the OS server release and diskless client release combination is supported. For more information, see “Preparing for Managing Diskless Clients”

(<http://docs.sun.com/app/docs/doc/817-1985/6mhm8o5me?a=view#clientconcept-28>).

Identify the diskless client platform by using this format:

```
instruction-set.machine-class.Solaris-version
```

For example:

```
i386.i86pc.Solaris_10
```

Possible platform options are shown in the following table.

<i>instruction-set</i>	<i>machine-class</i>	<i>Solaris-version</i>
sparc	sun4v sun4u sun4m sun4d sun4c	Solaris_10 (starting with the Solaris 10 1/06 OS) Solaris_9 Solaris_8 Solaris_7 Solaris_2.7
i386	i86pc	Solaris_10 Solaris_9 Solaris_8 Solaris_2.7

Note –

The sun-4c architecture is not supported in the Solaris 8, Solaris 9, or Solaris 10 releases. The sun-4d architecture is not supported in the Solaris 9 and Solaris 10 releases. The sun-4m architecture is not supported in the Solaris 10 release.

3. Identify the media path.

The media path is the full path to the disk image that contains the OS that you want to install for the diskless client.

The Solaris OS is delivered on multiple CDs. However, you cannot use the `smosservice` command to load OS services from a multiple-CD distribution. You must run the scripts that are found on the Solaris software CDs (and optional Language CD), as explained in the following steps.

4. Create an installation image on a server. For information on setting up an installation server, refer to *Solaris 10 Installation Guide: Network-Based Installations* (<http://docs.sun.com/app/docs/doc/817-5504>).

5. Load the required OS services from the CD image by using one of the following scripts:

- CD 1 – `/cdrom/cdrom0/s2/Solaris_10/Tools/setup_install_server`
- Additional Solaris Software CDs –
`/cdrom/cdrom0/s2/Solaris_10/Tools/add_to_install_server`
- Language CD –
`/cdrom/cdrom0/s2/Solaris_10/Tools/add_to_install_server`

For example, if you are using the `setup_install_server` script from the Solaris 10 Software 1 CD on a locally connected CD-ROM device, the syntax looks similar to the following.

```
# mkdir /export/install/sol_10_x86
# cd /cd_mount_point/Solaris_10/Tools
# ./setup_install_server /export/install/sol_10_x86
```

6. After the Solaris CD image is installed on the disk, note the disk media path. For example:

```
/net/export/install/sol_10_x86
```

This is the disk media path that needs to be specified when you use the `smosservice` command.

7. Identify the `SUNWCXall` cluster when you add OS services.

You must use the same cluster for diskless clients that run the same OS on the same system.

Adding OS Services for Diskless Client Support

Use this procedure to add OS services for a diskless client on the server.

Note –

When adding OS services with the `smosservice add` command, root (/) and /usr packages with the `ARCH=all` type are not installed. These packages are skipped. No warning or error messages are displayed. After you add the OS services to the OS server, you must install the missing packages manually. Use the information in section C of the Appendix to find and then add missing packages.

-
1. Become superuser or assume an equivalent role.

Roles contain authorizations and privileged commands. For more information about roles, see the *System Administration Guide: Security Services* (<http://docs.sun.com/app/docs/doc/816-4557/6maosrjfi?a=view>).

2. Verify that the Solaris Management Console server is running and that the diskless client tools are available on the system.

```
# /usr/sadm/bin/smosservice list -H host-name:898 --
```

3. Add the OS services.

```
# /usr/sadm/bin/smosservice add -H host-name:898 -- -o host-name
-x mediapath=path -x platform=instruction-set.machine-class.Solaris-version
-x cluster=cluster-name -x locale=locale-name
```

For example:

```
# /usr/sadm/bin/smosservice add -H test-serv:898 -- -o test-serv
-x mediapath=/export/install/x86_10 -x platform=i386.i86pc.Solaris_10
-x cluster=SUNWCXall
Authenticating as user: root

Type /? for help, pressing enter accepts the default denoted by [ ]
Please enter a string value for: password :: xxxxxx
Loading Tool: com.sun.admin.osservmgr.cli.OsServerMgrCli from varun:898
Login to varun as user root was successful.
Download of com.sun.admin.osservmgr.cli.OsServerMgrCli from test-serv:898 was
successful.
```

Adding a Diskless Client Boot Environment

Starting with the Solaris 10 1/06 release, use this procedure to add a diskless client after you have added OS services.

1. Become superuser or assume an equivalent role.

Roles contain authorizations and privileged commands. For more information about roles, see the *System Administration Guide: Security Services* (<http://docs.sun.com/app/docs/doc/816-4557/6maosrjfi?a=view>).

2. Add the diskless client.

```
# /usr/sadm/bin/smdiskless add -- -i ip-address -e ethernet-address
-n client-name -x os=instruction-set.machine-class.Solaris-version
-x root=/export/root/client-name -x swap=/export/swap/client-name
-x swapsize=size -x tz=time-zone -x locale=locale-name
```

For example:

```
# /usr/sadm/bin/smdiskless add -- -i 10.1.20.5 -e 0:0c:f1:a8:2c:da
-n varun -x os=i386.i86pc.Solaris_10 -x root=/export/home/root/varun
```

```
-x swap=/export/home/swap/varun -x swappsize=512
```

```
Loading Tool: com.sun.admin.ossvermgr.cli.OsServerMgrCli from varun  
Login to test-serv as user root was successful.  
Download of com.sun.admin.ossvermgr.cli.OsServerMgrCli from test-serv was successful.
```

The following files and directories are created in the /tftpboot directory:

```
drwxr-xr-x  6 root sys      512 Dec 28 14:53 client-host-name  
lrwxrwxrwx  1 root root      31 Dec 28 14:53 menu.lst.01ethernet-address  
-> /tftpboot/client-host-name/grub/menu.lst  
-rw-r--r--  1 root root 118672 Dec 28 14:53 01ethernet-address
```

If the console is on a serial port, edit the /tftpboot/menu.lst.01ethernet-address file. Uncomment the line that specifies the tty setting.

To change the default menu.lst file that is created on the client, edit the echo lines in the /usr/sadm/lib/wbem/config_tftp file.

Note -

Sometimes the /tftpboot directory is not populated properly. If the /tftpboot directory is not populated, first delete the entire /tftpboot directory (**rm -rf /tftpboot**) and then use the modified config_tftp script mentioned in section B of the Appendix to recreate /tftpboot as shown in the following example:

```
config_tftp add diskless-client-name diskless-client-mac-address
```

For example:

```
config_tftp add varun 0:0c:f1:a8:2c:da
```

-
3. If the DHCP options are not already created, add the BootSrvA and BootFile DHCP options to your DHCP server configuration to enable a PXE boot.

For example:

```
Boot server IP (BootSrvA) : svr-addr  
Boot file (BootFile) : 01client-macro
```

Here *svr-addr* is the IP address of the server and *client-macro* is named by the client's Ethernet type (01) and the MAC address of the client. This number is also the name of the file that is used in the `/tftpboot` directory on the installation server.

Note –

The *client-macro* notation consists of uppercase letters. The notation should not contain any colons.

For more information, see the following:

- “x86: How to Perform a GRUB Based Boot From the Network”
(<http://docs.sun.com/app/docs/doc/817-1985/6mhm8o5pq?a=view>)
 - “Preconfiguring System Configuration Information With the DHCP Service (Tasks)” in the *Solaris 10 Installation Guide: Network-Based Installations*
(<http://docs.sun.com/app/docs/doc/817-5504/6mkv4nh33?a=view>)
 - “Part III DHCP,” in the *System Administration Guide: IP Services*
(<http://docs.sun.com/app/docs/doc/816-4554/6maoq01r1?a=view>)
-

4. Make sure the `tftp` service is running on the TFTP server. If the service is not running, follow the steps mentioned in section D of the Appendix.
5. Boot the client (x86) using network PXE boot, which should boot up the client with the Solaris 10 x86 OS.

Distributing the TFTP/DHCP and NFS/Root Environment

Having both the TFTP and root environment in the same server may cause an overhead on the server if multiple diskless boot clients are to be set up. To reduce the load, you can distribute the TFTP/DHCP and NFS/root environments on separate servers. In order to keep the TFTP server separate from the root server, the process requires manually transferring the `tftpboot` contents to the `tftpboot` servers.

1. For a distributed TFTP/DHCP and NFS/root server setup, create a tar of the `/tftpboot` directory.

```
tar cvf tftpboot.tar /tftpboot
```
2. Disable DHCP services from the existing DHCP server and unmount the `/tftpboot` directory.
3. Transfer the tar file you created to the new TFTP/DHCP server and extract the TFTP tar file at the root directory.
4. Copy the `/etc/bootparam` and `/etc/ethers` files to the TFTP server.
5. Update the `/etc/hosts` file for the NFS server (`test-serv`, in this example).
6. Update the `/etc/vfstab` file for sharing the `/tftpboot` directory and mount.

7. Set the `BootSrvA` value to the IP address of the TFTP/DHCP server and the `BootFile` value to the `01client-mac-address` value obtained previously (for example, 01000CF1A82CDA).
8. Make sure TFTP services are running on the new TFTP/DHCP server. If necessary, follow section D of the Appendix to enable TFTP services.

Setting Up the DHCP Server

The following screen shots show an example DHCP setup.

The client ID is the tftpboot client's network ID (Ethernet ID). The macro name for the PXE client is `PXEClient:Arch:00000:UNDI:002001`. For diskless boot setup, it is mandatory to have a permanent IP allocation to the client.

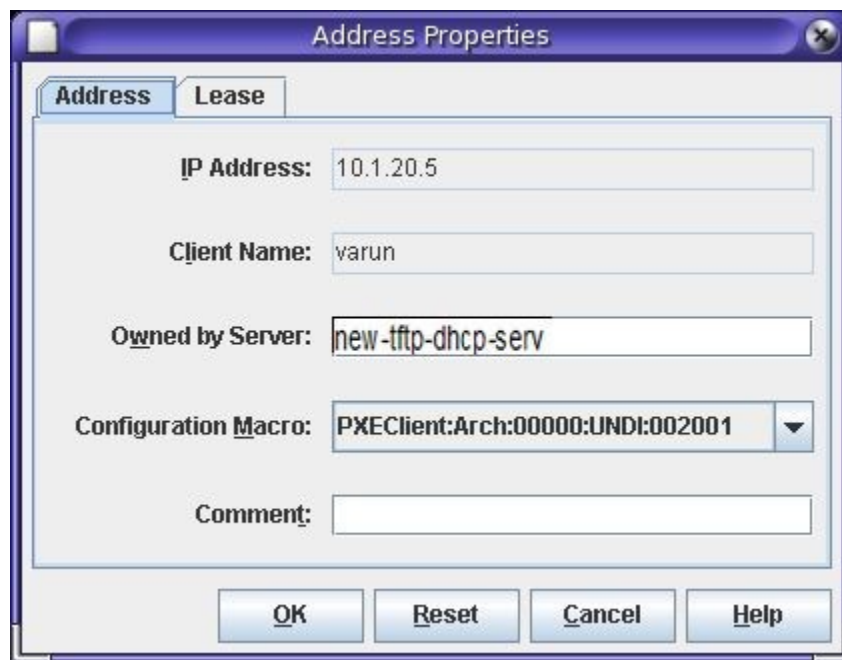


Figure 1: DHCP Address Properties

Note -

Dynamic IP allocation is not supported for diskless boot.

As shown in Figure 2, the example server `new-tftp-dhcp-serv` is the TFTP/DHCP server and its settings for DHCP ensure that the server IP address doesn't expire and it is a static IP address to the client.

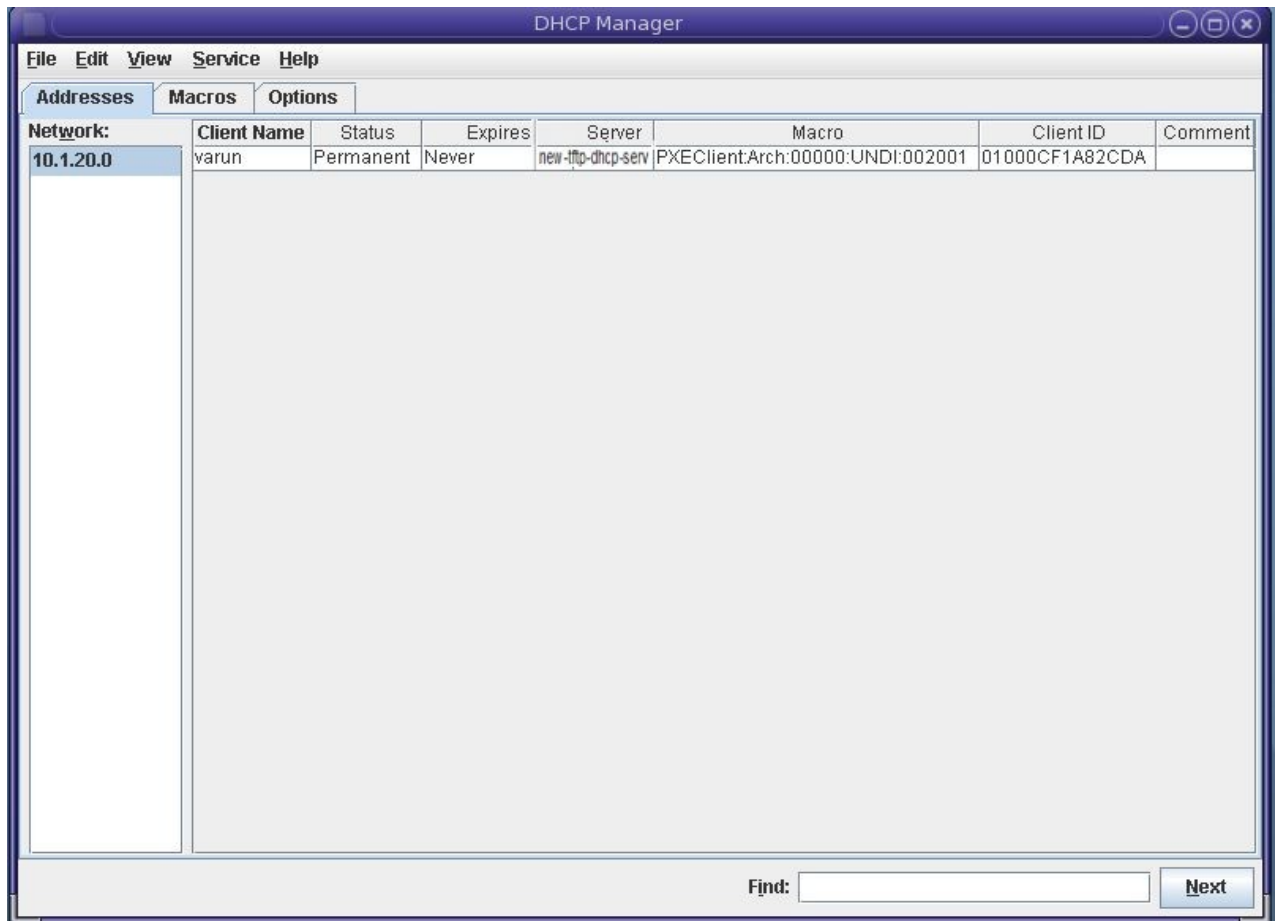


Figure 2: First Look at DHCP Settings

Figure 3 shows the vendor-based settings. Make sure that the classes under Vendor Client Classes are defined correctly and that they are in the right order. Here, the PXE boot client is `PXEClient:Arch:00000:UNDI:002001`.

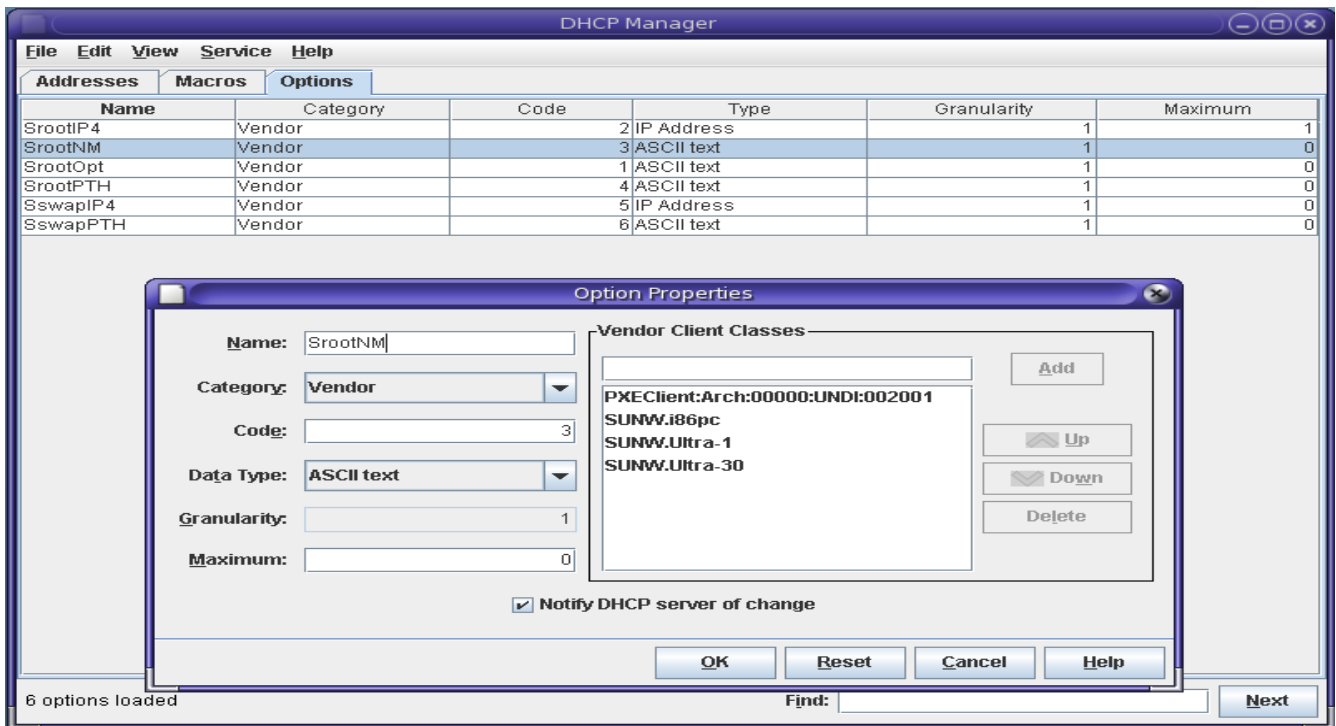


Figure 3: Vendor-Based DHCP Settings

Figure 4 shows the macro settings for the configuration settings required for DHCP. In this configuration, the location of the boot server and the boot file (tftpboot file) are defined along with vendor-based settings. Important settings are mentioned below with example values.

SrootIP4 (IP address of NFS root directory)	10.1.20.1
SrootNM (host name of root/NFS server)	testserv
SrootPATH (root path location)	/export/root/varun
SswapIP4 (swap server location)	10.1.20.1
SswapPTH (path to swap device)	/export/home/swap/varun

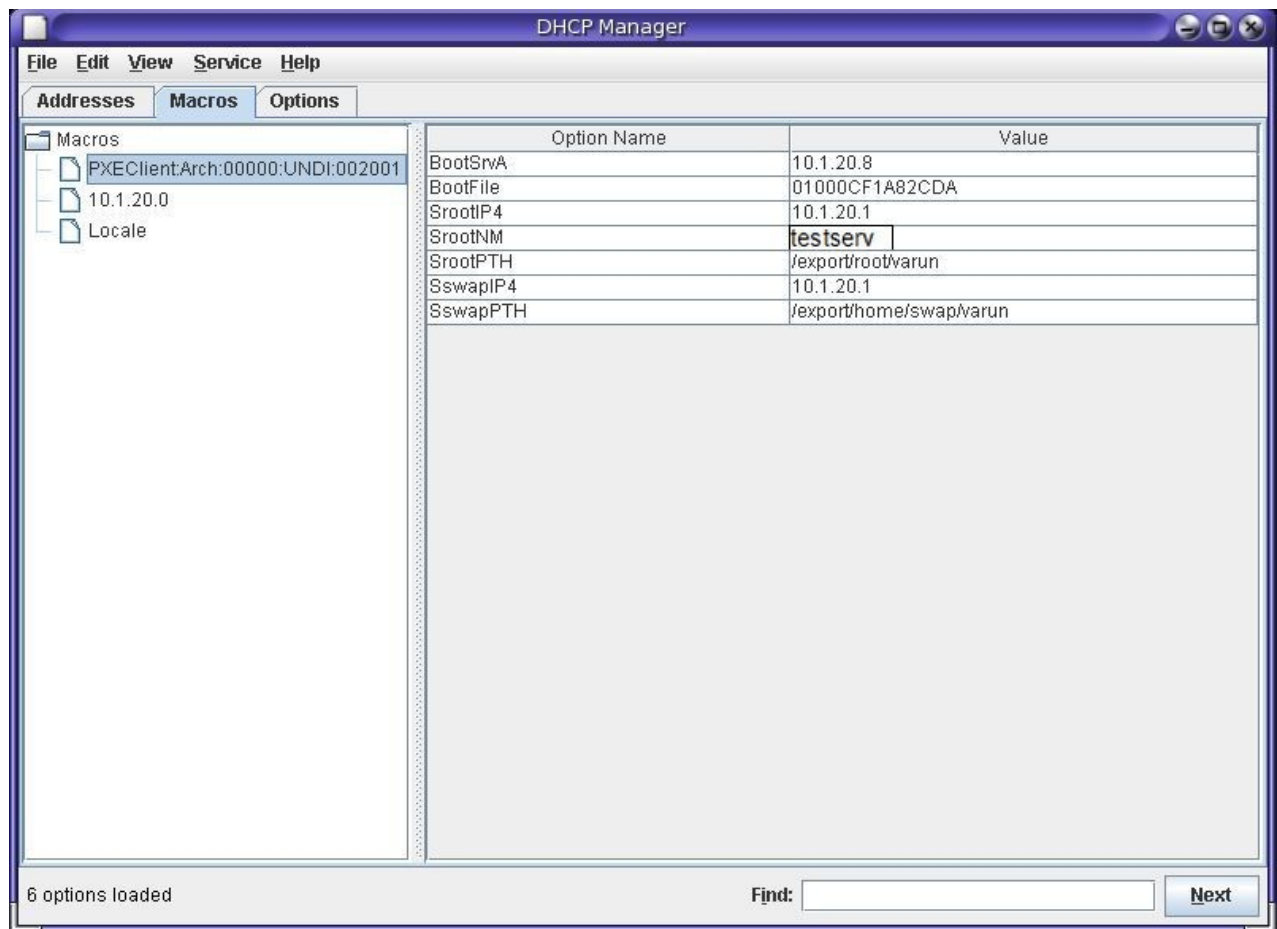


Figure 4: DHCP Macro Settings

Resolving Issues

Use the following information to resolve issues:

- Problem: Failure occurs when logging in to the Sun Java™ Desktop System (GNOME).
- Solution: Refer to the following weblogs:
 - Blog -- Diskless Solaris x86 Part 1: Setup Diskless Solaris Environment: http://blogs.sun.com/thaniwa/entry/en_diskless_solaris_x86_p1
 - Blog -- Diskless Solaris x86 Part 2: Use JDS (GNOME) on diskless client: http://blogs.sun.com/thaniwa/entry/en_diskless_solaris_x86_p2
- Problem: The root password is empty in the default settings of the diskless client.
- Solution: By default, `smosservice` does not set the root password of the diskless client. So the root password is empty at the time of the first boot. After setting up the diskless client OS by using the `smosservice` command, change the shadow password file of the diskless client by updating the `/export/root/varun/etc/shadow` file for assigning the password to the root account.

- Problem: Diskless client reports owner of the module `/usr/lib/security/pam_unix_session.so.1` is not root. Login attempt shows that the `/usr` file system is owned by nobody.
- Solution:
 - Using a text editor, modify this file on the diskless client's server:
`/export/root/client/etc/default/nfs`
 - Change the `#NFSMAPID_DOMAIN=domain` line to this:
`NFSMAPID_DOMAIN=the_same_value_as_in_server's_/var/run/nfs4_domain`
 - Ensure that the OS server and the diskless client have the same `nfsmapid` domain. To verify this information, check the `/var/run/nfs4_domain` file.
 - Reboot the diskless client.

Running Diagnosis Commands

Run the following diagnosis commands to check the traffic on the DHCP/TFTP server.

```
snoop -d iprb0 -o /tmp/snoop.output -v
snoop -i /tmp/snoop.output1 -x0 -v
```

To run the DHCP service in diagnostic mode, use the following command:

```
/usr/lib/inet/in.dhcpd -dv
```

References

- Managing Diskless Clients (Tasks):
<http://docs.sun.com/app/docs/doc/817-1985/clientconcept-33212?a=view>
- DHCP FAQ:
http://www.dhcp-handbook.com/dhcp_faq.html
- Blog -- Diskless Solaris x86 Part 1: Setup Diskless Solaris Environment:
http://blogs.sun.com/thaniwa/entry/en_diskless_solaris_x86_p1
- Blog -- Diskless Solaris x86 Part 2: Use JDS (GNOME) on diskless client:
http://blogs.sun.com/thaniwa/entry/en_diskless_solaris_x86_p2
- Synopsis: PXE Access Violation Before Primary Bootstrap Has Loaded:
http://docs.sun.com/source/817-5625-10/Claymore_Troubleshooting.html#20195

- Preconfiguring System Configuration Information With the DHCP Service (Tasks): <http://docs.sun.com/app/docs/doc/817-5504/6mkv4nh33?a=view>
- Troubleshooting Diskless Client Problems: <http://docs.sun.com/app/docs/doc/817-1985/qcfqn?a=view>

For More Information

Here are additional resources:

- Solaris downloads: <http://www.sun.com/download/index.jsp?cat=Operating%20Systems&tab=3&subcat=Solaris%20Operating%20Systems>
- Sun training courses: <http://www.sun.com/training/>
- Support:
 - Sun Services and Solutions: <http://sun.com/solutions>
 - SunSolve Online: <http://sunsolve.sun.com>
- Open source products and information: <http://www.sun.com/software/opensource/learnmore.jsp>
- Discussions, such as the Sun developer forums: <http://forum.java.sun.com/index.jspa>
- Wikis:
 - Wikis on sun.com: <http://wiki.sun.com/dashboard.action>
 - BigAdmin wiki: <http://wiki.sun.com/display/BigAdmin/Home>

About the Author

Vijay Upreti is working as an ISV engineer at Sun Microsystems in Bangalore, India. He has more than 12 years of UNIX System Administration experience.

Acknowledgements

The author would like to thank Rangarajan Vasudevan and M. M. Reddy of the ISV Support organization at Sun Microsystems.

Appendix

A. Example dhcp_config Script

```
# Load the Solaris vendor specific options. We'll start out supporting
# the Sun-Blade-1000, Sun-Fire-880, and i86 platforms. Note that the
# SUNW.i86pc option only applies for the Solaris 10 3/05 release.
# Changing -A to -M would replace the current values, rather than add them.
dhtadm -A -s SrootOpt -d \
'Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,1,ASCII,1,0'
dhtadm -A -s SrootIP4 -d \
'Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,2,IP,1,1'
dhtadm -A -s SrootNM -d \
'Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,3,ASCII,1,0'
dhtadm -A -s SrootPTH -d \
'Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,4,ASCII,1,0'
dhtadm -A -s SswapIP4 -d \
'Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,5,IP,1,0'
dhtadm -A -s SswapPTH -d \
'Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,6,ASCII,1,0'
dhtadm -A -s SbootFIL -d \
'Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,7,ASCII,1,0'
dhtadm -A -s Stz -d \
'Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,8,ASCII,1,0'
dhtadm -A -s SbootRS -d \
'Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,9,NUMBER,2,1'
dhtadm -A -s SinstIP4 -d \
'Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,10,IP,1,1'
dhtadm -A -s SinstNM -d \
'Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,11,ASCII,1,0'
dhtadm -A -s SinstPTH -d \
'Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,12,ASCII,1,0'
dhtadm -A -s SsysidCF -d \
'Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,13,ASCII,1,0'
dhtadm -A -s SjumpsCF -d \
'Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,14,ASCII,1,0'
dhtadm -A -s Sterm -d \
'Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,15,ASCII,1,0'
dhtadm -A -s SbootURI -d \
'Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,16,ASCII,1,0'
dhtadm -A -s SHTTPproxy -d \
'Vendor=SUNW.Sun-Blade-1000 SUNW.Sun-Fire-880 SUNW.i86pc,17,ASCII,1,0'
# Load some useful Macro definitions.
# Define all Solaris-generic options under this macro named Solaris.
dhtadm -A -m Solaris -d \
':SrootIP4=10.21.0.2:SrootNM="blue2":SinstIP4=10.21.0.2:SinstNM="red5":'
# Define all sparc-platform specific options under this macro named sparc.
dhtadm -A -m sparc -d \
':SrootPTH="/export/sparc/root":SinstPTH="/export/sparc/install":'
# Define all sun4u architecture-specific options under this macro named sun4u.
# (Includes Solaris and sparc macros.)
dhtadm -A -m sun4u -d ':Include=Solaris:Include=sparc:'
# Solaris on IA32-platform-specific parameters are under this macro named i86pc.
# Note that this macro applies only for the Solaris 10 3/05 release.
dhtadm -A -m i86pc -d \
':Include=Solaris:SrootPTH="/export/i86pc/root":SinstPTH="/export/i86pc/install"\
:SbootFIL="/platform/i86pc/kernel/unix":'
# Solaris on IA32 machines are identified by the "SUNW.i86pc" class. All
# clients identifying themselves as members of this class will see these
# parameters in the macro called SUNW.i86pc, which includes the i86pc macro.
# Note that this class only applies for the Solaris 10 3/05 release.
dhtadm -A -m SUNW.i86pc -d ':Include=i86pc:'
# Sun-Blade-1000 platforms identify themselves as part of the
# "SUNW.Sun-Blade-1000" class.
```

```

# All clients identifying themselves as members of this class
# will see these parameters.
dhtadm -A -m SUNW.Sun-Blade-1000 -d \
':SbootFIL="/platform/sun4u/kernel/sparcv9/unix":\
Include=sun4u:'
# Sun-Fire-880 platforms identify themselves as part of the "SUNW.Sun-Fire-880" class.
# All clients identifying themselves as members of this class will see these parameters.
dhtadm -A -m SUNW.Sun-Fire-880 -d \
':SbootFIL="/platform/sun4u/kernel/sparcv9/unix":Include=sun4u:'
# Add our boot server IP to each of the network macros for our topology served by our
# DHCP server. Our boot server happens to be the same machine running our DHCP server.
dhtadm -M -m 10.20.64.64 -e BootSrvA=10.21.0.2
dhtadm -M -m 10.20.64.0 -e BootSrvA=10.21.0.2
dhtadm -M -m 10.20.64.128 -e BootSrvA=10.21.0.2
dhtadm -M -m 10.21.0.0 -e BootSrvA=10.21.0.2
dhtadm -M -m 10.22.0.0 -e BootSrvA=10.21.0.2
# Make sure we return host names to our clients.
dhtadm -M -m DHCP-servername -e Hostname= NULL_VALUE
# Create a macro for PXE clients that want to boot from our boot server.
# Note that this macro applies for the Solaris 10 3/05 release.
dhtadm -A -m PXEClient:Arch:00000:UNDI:002001 -d \
:BootFile=nbp.i86pc:BootSrvA=10.21.0.2:
# Create a macro for PXE clients that want to boot from our boot server.
# Note that this macro applies for the Solaris 10 2/06 release.
dhtadm -A -m PXEClient:Arch:00000:UNDI:002001 -d \
:BootFile=i86pc:BootSrvA=10.21.0.2:
# Create a macro for the x86 based client with the Ethernet address 00:07:e9:04:4a:bf
# to install from the network by using PXE.
dhtadm -A -m 010007E9044ABF -d :BootFile=010007E9044ABF:BootSrvA=10.21.0.2:
# The client with this MAC address is a diskless client. Override the root settings
# which at the network scope setup for Install with our client's root directory.
dhtadm -A -m 0800201AC25E -d \
':SrootIP4=10.23.128.2:SrootNM="orange-svr-2":SrootPTH="/export/root/10.23.128.12":'

```

B. config_tftp File

```

#!/usr/bin/sh

#
# Copyright 2005 Sun Microsystems, Inc. All rights reserved.
# Use is subject to license terms.
#
#pragma ident"@(#)config_tftp.sh 1.1 05/09/06 SMI"
#
# This script is always invoked by libsmoss with three args:
# <subcmd> <clientname> <client ether address>
# The subcmd must be one of add, delete, and modify.
# The ether address is assumed to be in the usual format of
# x:x:x:x:x:x
# where x is a one or two digit hex number.
#
#
# Convert ETHERADDR to canonical form with hex digits in upper case
#
convert_etheraddr()
{
    ether_addr=
    for i in 1 2 3 4 5 6; do
        hex=`echo ${ETHERADDR} | cut -d : -f $i | tr '[a-f]' '[A-F]`
        ether_addr=${ether_addr}:`echo "ibase = 16 ; $hex" | bc`
    done
    ether_addr=`echo ${ether_addr} | \
awk -F: '{printf "%02x%02x%02x%02x%02x%02x", $2, $3, $4, $5, $6, $7}`
    ETHER_UPPER=`echo ${ether_addr} | tr '[a-f]' '[A-F]`
}

```

```

}

name_to_ipaddr()
{
    line=`grep -v "^#" /etc/hosts | grep "[ ]$1[ ]"`
    hostip=`echo $line | (read hostip junk junk junk; echo $hostip)`
    if [ X"$hostip" != X ]; then
        echo "$hostip"
        return 0
    fi
    echo "ipaddr-for-$1"
    return 0
}

#
# Modify client's bootenv.rc, build grub menu, create boot archive,
# and lofs mount client's /boot area under /tftpboot/<clientname>.
#
add_client()
{
    # lofs mount /boot of client to /tftpboot
    mkdir -p /tftpboot/${CLIENTNAME}
    mount -F lofs /export/root/${CLIENTNAME}/boot /tftpboot/${CLIENTNAME}
    echo "/export/root/${CLIENTNAME}/boot - /tftpboot/${CLIENTNAME} lofs - yes ro" >>
/etc/vfstab

    #
    # setup properties in bootenv.rc
    # Note: rootopts should be set to read only to avoid failure of
    # SMF boot-archive service.
    #
    BOOTENVRC=/export/root/${CLIENTNAME}/boot/solaris/bootenv.rc
    echo "setprop fstype 'nfsdyn'" >> ${BOOTENVRC}
    echo "setprop server-name '${HOSTNAME}'" >> ${BOOTENVRC}
    echo "setprop server-path '/export/root/${CLIENTNAME}'" >> ${BOOTENVRC}
    echo "setprop server-rootopts 'ro'" >> ${BOOTENVRC}

    # create boot archive link to /boot to make them available for tftp
    /sbin/bootadm -a update -R /export/root/${CLIENTNAME} 2> /dev/null
    rm -f /export/root/${CLIENTNAME}/boot/boot_archive
    ln /export/root/${CLIENTNAME}/platform/i86pc/boot_archive \
        /export/root/${CLIENTNAME}/boot/boot_archive

    if [ ! -f /export/root/${CLIENTNAME}/boot/multiboot ] ; then
        ln /export/root/${CLIENTNAME}/platform/i86pc/multiboot \
/export/root/${CLIENTNAME}/boot/multiboot
    fi

    # setup menu.lst file content
    menufile=/export/root/${CLIENTNAME}/boot/grub/menu.lst
    rm -f ${menufile}
    touch ${menufile}
    echo "default=0" >> ${menufile}
    echo "timeout=10" >> ${menufile}
    echo "title Solaris Diskless Client" >> ${menufile}
    echo " root (nd)" >> ${menufile}
    echo "# If console is on ttya|ttyb, replace kernel line with" >> ${menufile}
    echo "# one of the commented lines" >> ${menufile}
    echo " kernel /${CLIENTNAME}/multiboot" >> ${menufile}
    echo "# kernel /${CLIENTNAME}/multiboot -B console=ttya" >> ${menufile}
    echo "# kernel /${CLIENTNAME}/multiboot -B console=ttyb" >> ${menufile}
    echo " module /${CLIENTNAME}/boot_archive" >> ${menufile}

    # setup menu.lst.01<ether_addr> link
    convert_etheraddr
    rm -f /tftpboot/menu.lst.01${ETHER_UPPER}
    cp /tftpboot/${CLIENTNAME}/grub/menu.lst /tftpboot/menu.lst.01${ETHER_UPPER}
}

```

```

# copy over pxegrub; don't do symlink -- pxegrub must be at top level
rm -f /tftpboot/01${ETHER_UPPER}
cp /tftpboot/${CLIENTNAME}/grub/pxegrub /tftpboot/01${ETHER_UPPER}

echo "\nIf not already configured, enable PXE boot by creating" > /dev/tty
echo "a macro which contains the following values:" > /dev/tty
echo " Boot server IP (BootSrvA) : `name_to_ipaddr ${HOSTNAME}`" > /dev/tty
echo " Boot file      (BootFile) : 01${ETHER_UPPER}" > /dev/tty
echo "\nIf console is on a serial port, edit /tftpboot/menu.lst.01${ETHER_UPPER}" >
/dev/tty
echo "(see comments in the file)." > /dev/tty
}

#
# The only thing to do is recreate client's boot archive.
#
modify_client()
{
    /sbin/bootadm -a update -R /export/root/${CLIENTNAME}
}

#
# Unmount /tftpboot/<hostname>, delete it from /etc/vfstab, delete
# pxegrub, and menu.lst.
#
delete_client()
{
    umount /tftpboot/${CLIENTNAME}
    rmdir /tftpboot/${CLIENTNAME}
    grep -v "[ ]*/tftpboot/${CLIENTNAME}[ ]*lofs[ ]*" \
        /etc/vfstab > /etc/vfstab.config_tftp
    if [ $? = 0 ] ; then
        mv /etc/vfstab /etc/vfstab-
        mv /etc/vfstab.config_tftp /etc/vfstab
    else
        rm -f /etc/vfstab.config_tftp
    fi

    # The ether addr is not provided on delete, we try to
    # figure it out in order to cleanup.
    ether_upper=`ls -l /tftpboot/menu.lst.* | \
        grep /tftpboot/${CLIENTNAME} | \
        cut -f3 -d. | (read ether junk; echo $ether)`
    if [ X"$ether_upper" != X ] ; then
        rm -f /tftpboot/menu.lst.$ether_upper
        rm -f /tftpboot/$ether_upper
    fi
}

SUBCMD=$1
CLIENTNAME=$2
ETHERADDR=$3
HOSTNAME=`/usr/bin/hostname`
ETHER_UPPER=

# test for pxegrub based boot
if [ ! -f /export/root/${CLIENTNAME}/platform/i86pc/multiboot ] ; then
    exit 0
fi

case $SUBCMD in
    add)    add_client ;;
    modify) modify_client ;;
    delete) delete_client ;;
    *)     /usr/bin/printf "usage: $0 add|modify|delete <host> <ether>\n"
           exit 1 ;;
esac

```

C. Finding and Adding Missing Packages

```
find ./ -name pkginfo -exec grep -w ARCH=all {} /dev/null \;  
grep -w SUNW_PKGTYPE ./SUNWjdmk-base/pkginfo ./SUNWjhdem/pkginfo ./SUNWjhdev/pkginfo  
./SUNWjhdoc/pkginfo ./SUNWjhrt/pkginfo ./SUNWmlibk/pkginfo  
grep -w BASEDIR ./SUNWjdmk-base/pkginfo ./SUNWjhdem/pkginfo ./SUNWjhdev/pkginfo  
./SUNWjhdoc/pkginfo ./SUNWjhrt/pkginfo ./SUNWmlibk/pkginfo  
pkginfo -R /export/Solaris_10 SUNWjdmk-base SUNWjhdem SUNWjhdev SUNWjhdoc SUNWjhrt  
SUNWmlibk  
pkgadd -R /export/Solaris_10 -a /tmp/admin_usr -d . SUNWjdmk-base  
pkgadd -R /export/Solaris_10 -a /tmp/admin_usr -d . SUNWjhdem  
pkgadd -R /export/Solaris_10 -a /tmp/admin_usr -d . SUNWjhdev  
pkgadd -R /export/Solaris_10 -a /tmp/admin_usr -d . SUNWjhdoc  
pkgadd -R /export/Solaris_10 -a /tmp/admin_usr -d . SUNWjhrt  
pkgadd -R /export/Solaris_10 -a /tmp/admin_opt -d . SUNWmlibk
```

D. Enabling the TFTP Service

1. Edit `/etc/inet/inetd.conf` and uncomment the TFTP entry.
2. Run `/usr/sbin/inetconv`:

```
/usr/sbin/inetconv
```

Licensing Information

Unless otherwise specified, the use of this software is authorized pursuant to the terms of the license found at http://www.sun.com/bigadmin/common/berkeley_license.html.