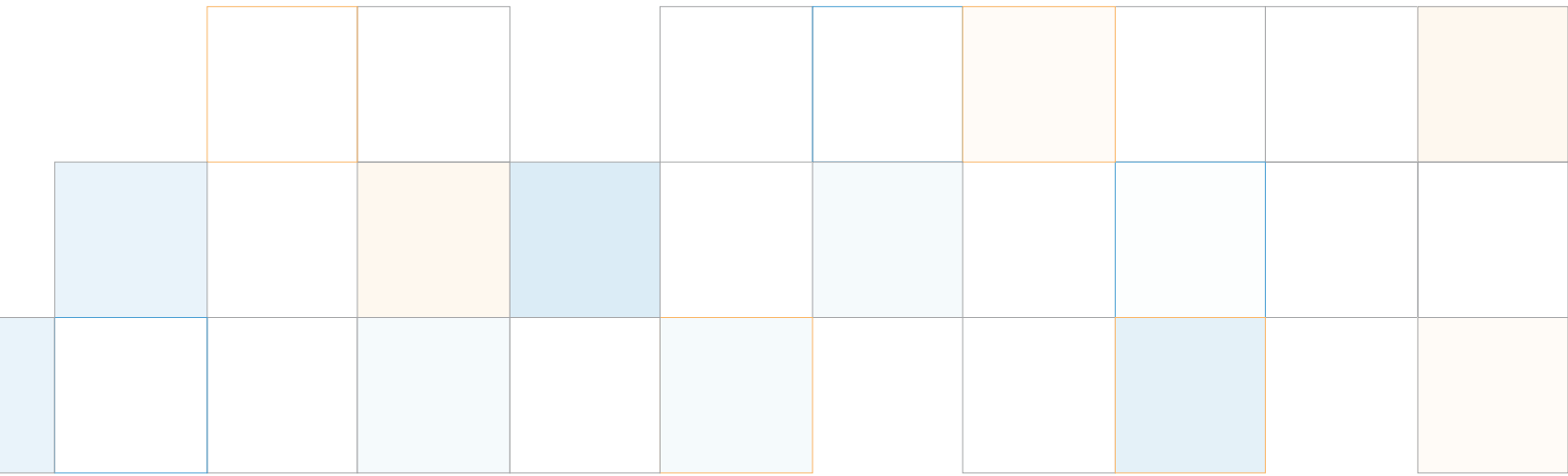




# tech docs



## How JumpStart Works

This Tech Tip is divided into the following specific sections.

- A) Overview of JumpStart software.
- B) How JumpStart works.
- C) Configuring the JumpStart server.
- D) Adding the client.
- E) Booting the client
- F) Listing of Files and Directories of interest.
- G) Troubleshooting
- H) Additional resources

## A) Overview of JumpStart software

### What is JumpStart Software?

JumpStart technology helps the system administrator manage complexity, by automating installation of Solaris[™] Operating System. It facilitates standardization of systems, enabling rapid deployment. JumpStart technology also minimizes opportunities for human error during system installation.

### Use JumpStart to:

- \* install new systems on the network.
- \* upgrade from one version of Solaris OS to another.
- \* re-install the Solaris environment on existing systems.

## B) How JumpStart works

### First Level Boot Process

- 1) When the "boot net - install" command is issued at the ok prompt, the JumpStart client looks for a Jumpstart boot server.
- 2) The boot server responds to the rarp request via the "rarpd" daemon (in.rarpd). Using the information in the /etc/ethers file, the server can obtain the client's IP address and communicate it to the client.
- 3) A JumpStart server, on the local subnet, receives the RARP request (Reverse Address Resolution Protocol), and maps it to an IP address, using its /etc/ethers and /etc/hosts files. A name service, such as NIS or NIS+, could also be used to map the address.
- 4) With the IP address known, the JumpStart server generates a RARP reply to the JumpStart client.
- 5) The JumpStart server responding to the client's RARP request maps the client's ethernet address to its IP address and host name, returning this data to the client.

## Second Level Boot Process

- 1) The JumpStart client downloads a minimal kernel(miniroot) from the JumpStart server, into the JumpStart client's memory. This comes by way of a TFTP request, issued by the Jumpstart client.
- 2) When the JumpStart server receives the TFTP request, it searches for a matching IP address and architecture, in the "/tftpboot" directory.
- 3) Once the JumpStart client is booted from the miniroot, it locates the "rules.ok" file. The entry is checked, to make sure it matches that of the JumpStart client.
- 4) When the match is found, the actions specified are executed. First, the "begin" scripts(if any) are executed. Then the specified profile is installed, and finally the "finish" scripts(if any) are executed.

### **\* A Note About Boot Servers!!!!!!**

Normally, The JumpStart server provides the boot program for booting clients. However, under one condition, the Solaris OS network booting architecture, requires you to set up a separate "boot server". A boot server, is a system with just enough information to boot up a client over a network. A boot server must be set up, when the install client is on a different sub-net from the install server.

SPARC[R] based technology install- clients, require a boot server when they exist on different subnets, because the network booting architecture uses the Reverse Address Resolution Protocol(RARP).

When a client boots, it issues a RARP request in order to obtain its IP address. RARP, however does not acquire the netmask number, which is required to distribute information across a router on a network. If the install/boot server exists across a router the boot will fail because the network traffic cannot be routed correctly without a netmask number. The result is, that you can install a client across a router, but you cannot boot a client across a router. So you will have to setup a separate boot server, on the same subnet as the client.

## C) Configuring the JumpStart server

This scenario has a system that is a boot, install, and configuration server all in one. The client is a SPARC system. It is based on the Solaris 9 OS.

### **Note!!!**

Keep in mind, that the "\"s after some of the command lines, depict a new line and are not actual parts of the command syntax.

- 1) Gather all system and network information.

```
boot/install/config server name SERVER
boot/install/config server IP address 129.151.29.1
OS image directory/export/install
configuration directory/Jumpstart
Client Information
Name CLIENT
Ethernet address 8:0:20:ab:cd:ef
IP address 129.151.29.10
Architecture sun4u
```

## 2) Create the boot/install server.

First, load the OS image from the Solaris 9 OS CD's onto the server's local disk. In this example, there will be another directory level(sol9-U2 - Solaris 9 Update 2 (12/02)) created, under "/export/install", to help define the version of the image installed. This helps when installing multiple versions of Solaris OS onto a single system.

### Note!!!

Solaris 2.6 and Solaris 7 OS, only have one Software CD. Solaris 8 OS and up, have two OS CD's(Solaris Software 1 of 2 and 2 of 2). This procedure will follow the installation using the 1 of 2 and 2 of 2 CD's.

Install the Software 1 of 2 CD.

Insert the Software 1 of 2 CD.

```
SERVER# cd /cdrom/cdrom0/s0/Solaris_9/Tools
SERVER# ls
Boot dial setup_install_server
add_install_client rm_install_client
SERVER# mkdir -p /export/install/sol9-U2
SERVER# ./setup_install_server /export/install/sol9-U2
Verifying target directory...
Calculating the required disk space for the Solaris_9 product
Calculating space required for the installation boot image
Copying the CD image to disk...
Copying Install Boot Image hierarchy...
Install Server setup complete
SERVER# cd /
SERVER# eject cdrom
```

Install the Software 2 of 2 CD.

Insert the Software 2 of 2 CD.

```
SERVER# cd /cdrom/cdrom0/Solaris_9/Tools
SERVER# ls
Installers add_to_install_server
SERVER# ./add_to_install_server /export/install/sol9-U2
The following Products will be copied to
/export/install/sol9-U2/Solaris_9/Product:
Solaris_2_of_2
```

```
If only a subset of products is needed enter Control-C
and invoke ./add_to_install_server with the -s option.
Checking required disk space...
/
Copying the Early Access products...
4124080 blocks
Copying Top Level installer...
1748208 blocks
Copying Tools Directory...
50976 blocks
Processing completed successfully.
SERVER#
```

### Note!!!

To setup a “boot server” only, run the following commands to create the boot image. Also, there is no need to run the “add\_to\_install\_server” command to add the 2 of 2 CD to the boot image.

```
SERVER# cd /cdrom/cdrom0/s0/Solaris_9/Tools
SERVER# ./setup_install_server -b /export/install/sol9-U2
```

### 3) Create the configuration directory on the server.

Now that the install server image has been installed, the configuration portion can be set up. These files are responsible for the automation of the installation. Create the directory, and copy the necessary files in order to perform a custom JumpStart installation. This is set up by copying the sample directory from the OS image directory(/export/install/...), to the /Jumpstart directory.

```
SERVER# mkdir /Jumpstart
SERVER# cd /export/install/sol9-U2/Solaris_9/Misc
SERVER# cp -r jumpstart_sample/* /Jumpstart
SERVER# cd /Jumpstart
SERVER# ls
any_machine net924_sun4c upgrade x86-class
check* rules x86-begin
host_class set_root_pw* x86-begin.conf/
SERVER#
```

### 4) Create a Profile for the system.

This file is used as a template for the custom Jumpstart installation. In this example, the default profile called “any\_machine” (listed below) will not be used. Instead, a new file called profile-9 will be created. This file contains the information to do an Initial install with the “entire distribution plus OEM” software cluster (SUNWCxall).

This file has added lines that will delete the power management software packages. It also supplies explicit disk partitioning parameters. The OS will be installed onto

one large slice(c0t0d0s0), with a second one (c0t0d0s1) used for swap. The swap slice is configured for 400MB's, with root taking the remaining space(free) of the disk.

```
SERVER# cat /Jumpstart/any_machine
install_type initial_install
system_type standalone
partitioning default
cluster SUNWCuser
cluster SUNWCxgl delete
package SUNWaudmo add
filesysany 40 swap
filesysany 50 /opt
SERVER# vi profile-9
install_type initial_install
system_type standalone
partitioning explicit
package SUNWpmowm delete
package SUNWpmowr delete
package SUNWpmowu delete
package SUNWpmm delete
package SUNWpmu delete
package SUNWpmux delete
cluster SUNWCXall
filesysc0t0d0s0 free /
filesysc0t0d0s1 400 swap
```

5) Create the sysidcfg file. The sysidcfg file is used to automate the system identification portion of the Solaris install. Following, is the one used for this installation.

### Note!!!

The minimum amount of information required to perform an automated JumpStart software installation is the following:

```
Host Name IP address system locale
time zone netmask IPv6
terminal type security policy name service
timeserver
SERVER# vi /Jumpstart/sysidcfg
system_locale=C
timezone=US/Eastern
timeserver=129.151.29.1
network_interface=primary {ip_address=129.151.29.10 \
protocol_ipv6=no netmask=255.255.255.0 default_route=NONE}
terminal=dtterm
root_password='8RVWDu1PHWhkY'
```

```
name_service=NONE
security_policy=NONE
```

### Note!!!

\*To use "name\_service=NONE" with Solaris 2.6 OS, patch 106193-03 or greater, must be loaded.

\*For Solaris 8 OS, the following parameters must be included:

```
network_interface=primary {protocol_ipv6=???} security
policy=???
```

\*For Solaris 9 OS, it may be necessary to include the dependent

```
keyword: {default_route=???
```

## 6) Update the Rules file.

The "rules" file is a text file used to create the "rules.ok" file, and is probably the most important file for custom Jumpstart installations. View this file as a look-up table consisting of one or more rules that define how install clients are installed, based on their system attributes. A rules file entry has 5 fields. The syntax of the rules file must follow the convention below. In this example, we will use the "any" keyword for the first rule (system attributes) and the file "any\_machine" for the fourth rule (profile name) and all others are left blank. ("- " = match always succeeds )

```
SERVER# cat /Jumpstart/rules
####
####
#
any - - any_machine -
^ ^ ^ ^ ^
| | | | |
| | | | |----- Finish script
| | | |----- Profile
| | |----- Begin script
| |----- Rule Value (specific system attribute)
|----- Rule keyword (general system attributes)
```

## 7) Check the rules file.

This is run to validate the rules file. This command creates the "rules.ok" file which is required by the installation software to match install clients to the predetermined rules.

```
SERVER# cd /Jumpstart
SERVER# ./check
Validating rules...
Validating profile any_machine...
The custom Jumpstart configuration is ok.
```

```
SERVER# cat rules.ok (check for any unwanted lines!!)
any - - any_machine -
SERVER#
```

8) Check to make sure the proper directories are shared.

Add the configuration (/Jumpstart) directory into the dfstab file.  
The following example is how the dfstab file should look.

### Note!!!

If the /Jumpstart entry doesn't exist then, this line will have to be added manually. Type in "shareall" to enable all the shared entries.

```
SERVER# cd /etc/dfs
SERVER# more dfstab
# Place share(1M) commands here for automatic execution
# on entering init state 3.
#
# Issue the command '/etc/init.d/nfs.server start' to run
# the NFS daemon processes and the share commands, after
# adding the very first entry to this file.
#
# share [-F fstype] [-o options] [-d "<text>"]
<pathname>[resource]
#
# .e.g,
# share -F nfs -o rw=engineering -d "home dirs"
/export/home2
share -F nfs -o ro,anon=0 /export/install
share -F nfs -o ro,anon=0 /Jumpstart
```

### D) Set up the client to install over the network

After setting up the /Jumpstart directory and appropriate files, the "add\_install\_client" command is used on the server, to setup the client to install Solaris from the server. Also, add the client into the "/etc/hosts" and the /etc/ethers files. This can be done in two ways: either, edit the files manually using "vi" as shown below, or use the "-i" and "-e" options within the add\_install\_client command (shown in the example).

```
SERVER# vi /etc/hosts
#
# Internet host table
#
```

```

127.0.0.1    localhost
129.151.29.1    SERVER loghost
129.151.29.10   CLIENT    <----- Add this line!
~
SERVER# vi /etc/ethers
8:0:20:ab:cd:ef    CLIENT
The proper syntax for the add_install_client command is:
# ./add_install_client -e <CLIENT_ETHERNET_ADDRESS>
-s <INSTALL_SERVER>:<OS_IMAGE_DIRECTORY>
-c <CONFIG_SERVER>:<CONFIGURATION_DIRECTORY>
-p <CONFIG_SERVER>:<PATH_TO_SYSIDCFG_FILE>
***-n [SERVER]:name_service[netmask]
CLIENT_NAME  ARCHITECTURE
*** Note!    (The brackets "[ ]" are needed!!!)
SERVER# cd /export/install/Solaris_2.6/Tools
SERVER# ls
Boot  dial  setup_install_server
add_install_client  rm_install_client
SERVER# ./add_install_client -e 8:0:20:ab:cd:ef \
-i 129.151.29.10 -s SERVER:/export/install \
-c SERVER:/Jumpstart -p SERVER:/Jumpstart CLIENT sun4u
Adding "share -F nfs -o ro,anon=0 /export/install" to
/etc/dfs/dfstab
making /tftpboot
enabling tftp in /etc/inetd.conf
updating /etc/bootparams
copying inetboot to /tftpboot

```

Breakdown of the "add\_install\_client" command...

- e Adds the clients info into the "/etc/ethers" file.
- i Adds the clients info into the "/etc/hosts" file.
- s Specifies the name of Install server (SERVER) and path (/export/install/Solaris\_2.6/) to the OS image This option is necessary if the client is being added to boot server.
- c Specifies the server (SERVER) and path (/Jumpstart) to locate the configuration files.
- p This specifies the configuration server (SERVER) and the path (/Jumpstart) to the "sysidcfg" file.
- n This option specifies which name service should be used during system configuration. This sets the "ns" keyword in the bootparams(4) file.  
name\_service Valid entries are "nis", "nisplus", and "none".  
SERVER - The name of the server or IP address of the specified name service.

If the server specified is on a different subnet, then the netmask may be needed to enable the client to contact the server.

- NETMASK - The netmask value specified in /etc/netmasks
- SERVER - Is the name of the boot/install/configuration server.
- CLIENT - Is the name of the JumpStart client.
- sun4u - Is the type of architecture for the client.

## E) Booting the client

Boot the client and install the Solaris OS software  
This is done at the client(CLIENT).

### Note!!!

Following, are the bootup and installation messages, as a guide to help understand this process better. There are also added comments(preceded by a “\*”), to show what is happening during certain intervals of the initial booting process.

ok boot net - install

\* Client sends RARP request to the boot server.

Boot device: /sbus/SUNW,hme@e,8c00000 File and args: - install  
24000 <---- (download of the tftpboot file.)

\* The server's "in.rarpd" daemon responds to the RARP request and  
\* delivers the clients IP address from info in the ethers and hosts  
\* files.

NOTICE: 64-bit OS installed, but the 32-bit OS is the default  
for the processor(s) on this system.  
See boot(1M) for more information.

Booting the 32-bit OS ...

\* The client, using "tftp", asks for the client stub. The server's  
\* inetd daemon listens for, and handles the request. The clients  
\* request then spawns the in.tftp daemon to handle this, and load  
\* the boot image. From the client, the boot image issues a  
\* "hostconfig" request for boot parameters(bootparams).  
\* Once the client has this info, the boot program on the client,  
\* mounts the root file system from the server, and  
\* "/kernel/genunix" starts the "init" program and loads the  
\* kernel.

SunOS Release 5.9 Version Generic\_112233-01 32-bit  
Copyright 1983-2002 Sun Microsystems, Inc. All rights reserved.  
Use is subject to license terms.

Booting the 32-bit OS ...

Configuring /dev and /devices

Using RPC Bootparams for network configuration information.

Configured interface hme0

Searching for configuration file(s)...

- \* The client searches for the configuration server, which was
- \* supplied by information from the bootparams file. The client
- \* mounts that directory and "sysidtool" is run. The client uses
- \* this information to start "suninstall" and eventually installs
- \* itself.

Using sysid configuration file 129.151.29.1:/Jumpstart/sysidcfg

Search complete.  
syslog service starting.  
savecore: no dump device configured  
Running in command line mode

Please wait while the system information is loaded... /

Please wait while the system is configured with your settings...

Generating software table of contents  
[this may take a few minutes...]  
Table of contents complete.  
Starting Solaris installation program...  
Searching for Jumpstart directory...  
Using rules.ok from 129.151.29.1:/Jumpstart.  
Checking rules.ok file...  
Using profile: profile-9  
Executing Jumpstart preinstall phase...  
Searching for SolStart directory...  
Checking rules.ok file...  
Using begin script: install\_begin  
Using finish script: patch\_finish  
Executing SolStart preinstall phase...  
Executing begin script "install\_begin"...  
Begin script install\_begin execution completed.  
Processing profile  
Selecting cluster (SUNWCXall)

#### Installing 64-bit Solaris packages

- Deselecting package (SUNWpmowm)
- Deselecting package (SUNWpmowr)
- Deselecting package (SUNWpmowu)
- Deselecting package (SUNWpmr)
- Deselecting package (SUNWpmu)
- Deselecting package (SUNWpmux)

**WARNING:** Unknown package ignored (SUNWaudmo)

- Selecting all disks
- Configuring boot device

- Configuring swap (c0t0d0s1)
- Configuring / (c0t0d0s0)
- Deselecting unmodified disk (c0t0d0)
- Deselecting unmodified disk (c0t1d0)
- Verifying disk configuration
- Verifying space allocation
- Total software size: 1000.03 Mbytes
- Preparing system for Solaris install
- Configuring disk (c0t0d0)
- Creating Solaris disk label (VTOC)
- Creating and checking UFS file systems
- Creating / (c0t0d0s0)

#### Beginning Solaris software installation

##### Starting software installation

```
SUNWcar.u...done.  990.49 Mbytes remaining.
FJSVvplu.u...done.  990.48 Mbytes remaining.
FJSVvplr.u...done.  990.47 Mbytes remaining.
SUNWctlu....done.  990.34 Mbytes remaining.
SUNWcsu.....done.  975.04 Mbytes remaining.
SUNWcsr.....done.  956.63 Mbytes remaining.
```

....

```
{Messages of package installation excluded}
```

....

```
SUNWnfssx...done.  1.46 Mbytes remaining.
SUNWsolnm...done.  1.45 Mbytes remaining.
```

##### Completed software installation

Solaris 9 software installation succeeded

##### Customizing system files

- Mount points table (/etc/vfstab)
- Unselected disk mount points  
(/var/sadm/system/data/vfstab.unselected)
- Network host addresses (/etc/hosts)

Cleaning devices

Customizing system devices

- Physical devices (/devices)
- Logical devices (/dev)

Installing boot information

- Installing boot blocks (c0t0d0s0)

Installation log location

- /a/var/sadm/system/logs/install\_log (before reboot)
- /var/sadm/system/logs/install\_log (after reboot)

Installation complete

Executing SolStart postinstall phase...

Executing finish script "patch\_finish"...

Finish script patch\_finish execution completed.

Executing Jumpstart postinstall phase...

The begin script log 'begin.log'

is located in /var/sadm/system/logs after reboot.

The finish script log 'finish.log'

is located in /var/sadm/system/logs after reboot.

syncing file systems... done

rebooting...

Resetting...

**F) A listing of some of the files and directories that are created or changed on the boot server(SERVER) during the installation procedure**

```
SERVER# cd /tftpboot
```

```
SERVER# ls -la
```

```
total 362
```

```
drwxrwxr-x 2 root other 512 Jan 16 13:10 ./
```

```
drwxr-xr-x 47 root root 1024 Jan 16 13:10 ../
```

```
lrwxrwxrwx 1 root other 26 Mar 11 12:08 8194C231.
```

```

SUN4U ->
inetboot.SUN4U.Solaris_9-1*
-rwxr-xr-x 1    root other      131328    May 30  2002
inetboot.SUN4U.Solaris_9-1*
-rw-r--r--      1    root other      301    Jan 16 13:10
rm.129.151.29.10
lrwxrwxrwx 1    root other      1    Jan 16 13:10 tftpboot ->./

SERVER# cat /etc/ethers
8:0:20:ab:cd:ef CLIENT

SERVER# cat /etc/hosts
127.0.0.1 localhost
129.151.29.1    SERVER    loghost
129.151.29.10  CLIENT

SERVER# cat /etc/bootparams

CLIENT root=SERVER:/export/install/sol9-U2/Solaris_9/Tools/Boot
\
install=SERVER:/export/install/sol9-U2 boottype=:in    \
sysid_config=SERVER:/Jumpstart    \
install_config=SERVER:/Jumpstart rootopts=:rsize=32768

```

### Note!!!

You may have to add NONE or the particular Name Service at the end of the bootparams line in order to overcome any problems loading over the network.

```

SERVER# cat /etc/dfs/dfstab
# Place share(1M) commands here for automatic execution
# on entering init state 3.
#
# Issue the command '/etc/init.d/nfs.server start' to run
# the NFS daemon processes and the share commands, after
# adding the very first entry to this file.
#
# share [-F fstype] [-o options] [-d "<text>"]
<pathname>[resource]
#      .e.g,
# share -F nfs -o rw=engineering -d "home dirs"
/export/home2
share -F nfs -o ro,anon=0 /export/i

SERVER# cat /etc/nsswitch.conf
#
# /etc/nsswitch.nis:

```

```

#
# An example file that could be copied over to /etc/nsswitch.
conf; it
# uses NIS (YP) in conjunction with files.
#
# "hosts:" and "services:" in this file are used only if the
# /etc/netconfig file has a "-" for nametoaddr_libs of "inet"
transports.
# the following two lines obviate the "+" entry in /etc/passwd
and
# /etc/group.
Passwd:    files nis
group:files nis

# consult /etc "files" only if nis is down.
Hosts:files nis
networks:  nis [NOTFOUND=return] files
protocols: nis [NOTFOUND=return] files
rpc:      nis [NOTFOUND=return] files
ethers:    files nis
netmasks: nis [NOTFOUND=return] files
bootparams:files nis
publickey: nis [NOTFOUND=return] files

netgroup:  nisnstall
share -F nfs -o ro,anon=0 /Jumpstart
automount: files nis
aliases:   files nis

# for efficient getservbyname() avoid nis
services:  files nis
sendmailvars:  files

SERVER# cd /Jumpstart

SERVER# ls -la
total 84
drwxr-xr-x 4   root other      512   Jan 16 12:58   .
drwxr-xr-x 47  root root 1024   Jan 16 13:10   ..
-rw-r-r-- 1   root other      182   Jan 16 08:22   any_machine
-rwxr-xr-x 1   root other     30029   Jan 15 13:34   check
drwxr-xr-x 3   root other      512   Jan 15 13:37   Jumpstart_
sample
-rw-r-r-- 1   root other     4742   Jan 15 13:34   rules
-rw-r-r-- 1   root other       54   Jan 16 08:22   rules.ok
-rw-r-r-- 1   root other     1622   Oct 16 10:56   profile-9
-rw-r-r-- 1   root other      188   Jan 15 13:36   sysidcfg

```

```
SERVER# cat sysidcfg
system_locale=C
timezone=US/Eastern
timeserver=129.151.29.1
network_interface=primary {ip_address=129.151.29.10 \
protocol_ipv6=no netmask=255.255.255.0 default_route=NONE}
terminal=dtterm
root_password='8RVWDu1PHWhkY'
name_service=NONE
security_policy=NONE
```

```
SERVER# cat rules.ok
any - - profile-9 -
# version=2 checksum=11112
```

```
SERVER# cat profile-9
install_type      initial_install
system_type       standalone
partitioning      explicit
package           SUNWpmowm delete
package           SUNWpmowr delete
package           SUNWpmowu delete
package           SUNWpmr delete
package           SUNWpmu delete
package           SUNWpmux delete
cluster           SUNWCXall
filesysc0t0d0s0  free /
filesysc0t0d0s1  400 swap
```

## G) Some Troubleshooting Hints

1) Use the snoop command. Reference the the manual pages for the command.  
A couple of options that work well for debugging are:

```
snoop -o snoop.out Client_Ethernet_Address
then run:
snoop -i snoop.out
```

The use of the Client\_Ethernet\_Address will eliminate any traffic not addressed to or from the client in question. Additionally, it will allow the viewing of the early portions of the boot process including the initial RARP and loading of the inetboot program.

You can just use snoop without naming a system, to capture all network activity (often more data than you'll need - although you can limit the protocols monitored).

Or, modify the "snoop all activity" approach, by naming one of the involved servers AND

the JumpStartClient, to monitor the activity or lack of it, between those two systems only.

Also, observe the boot messages for clues as to what may be happening.

## 2) ARP/RARP errors while trying to boot up the client.

This Error occurs when you boot an install client over the network, but the install client can't locate the boot server. This is usually caused by insufficient or incorrect information in the hosts and or ethers files. Also the "in.rarpd" daemon may not be running. Check to make sure /etc/nsswitch.conf file indicates the correct search path (i.e. ethers files nis). Another daemon that can cause a client not to boot is "in.tftpd". Make sure this is running (# ps -eflgrep tftpd). If it's not running, then check the file "/etc/inetd.conf" and make sure the following line is not commented out: (No # symbol preceding this line) "tftp dgram udp wait root /usr/sbin/intftpd intftpd -s /tftpboot"

## 3) le0, hme0, eri0, etc...: no carrier errors.

This is caused by the system not being connected to the network, or there is a problem with the network port.

## 4) RPC: Timed out

This error occurs while trying to boot over the network, and encountering problems with the bootparams file. Check the file for the proper information and spelling. You may have to end up removing and re-adding the client with rm\_install\_client and add\_install\_client.

## 5) Searching for Jumpstart directory...not found

The install client is booting, but it fails to access the Jumpstart directory. This will cause the "Hands Off" install to go interactive. This can be caused by a couple of things. An incorrect path name specified in the add\_install\_client -c command, or the Jumpstart directory isn't shared properly.

## 6) WARNING: getfile: RPC failed: error 5 (RPC Timed out)

This is usually a result of two or more servers responding to the same boot request. The install client may attach itself to the wrong server, that contains the wrong information.

Possible areas to look at, are the bootparams file(look for the correct boot server), or the /tftpboot directory may have multiple entries for the same client

## 7) Error: "warning: could not find matching rules in rules.ok file"

Or it goes straight into an Interactive install.

## **Problem:**

This occurs when the client can't access the Jumpstart directory. An interactive installation will occur with this error. There are multiple reasons for this message to occur.

- A) If you specified an incorrect path to the Jumpstart directory after running the command "add\_install\_client".
- B) The Jumpstart directory wasn't shared properly on the install server, or the permissions are incorrect.
- C) The rules/rules.ok file(s) may have problems.
- D) A mismatch within the netmask entries.
- E) The Jumpstart directory is bypassed.

### **How to fix the problem:**

#### **Solution for problem A:**

Remove and re-add the client using the correct path to the Jumpstart directory.

#### **Solution for problem B:**

Check the /etc/dfs/dfstab file to make sure the Jumpstart directory is shared correctly. (i.e. share -F nfs -o ro,anon=0 /Jumpstart)

Make sure the permissions for the Jumpstart directory are set to 755 (drwxr-xr-x).

#### **Solution for problem C:**

As stated in SRDB 16678 - remove the rule keyword 'network' from the rules file, run check again, and then Jumpstart should run ok. This is a known bug: Bug Id: 1259120. Also there may be syntax problems, or keywords not in the correct order.

#### **Solution for problem D:**

The netmask values may not match between the "/etc/netmasks" and the "OS image/netmask" (OS\_IMAGE/Tools/Boot/netmask) files.

The ifconfig command should also be run, to verify that the netmasks entry is correct. The following example, shows how the files should appear. Also, check the sysidcfg file (see the example below). There may be an entry for the "network\_interface" that doesn't match the /etc/netmasks, ifconfig, and OS\_IMAGE/Tools/Boot/netmask values.

#### **Solution for problem E:**

The Solution here, is to make sure you type in "boot net - install". It could be, that the " - install" is missing from the "boot net" command. Not using the full command will bypass the Jumpstart directory, and go right into an Interactive install.

## H) Other resources to check out

### Reference Manuals

- \* Solaris Advanced Installation Guide (Solaris 8) - p/n 806-7932-10
- \* Solaris Installation Guide (Solaris 9) - p/n 816-5102-10
- \* Automating Solaris Installations (A Custom Jumpstart Guide) - isbn 0-13-312505-X
- \* Jumpstart Technology (Sun Blueprints) - isbn 0-13-062154-4

### SRDB's

- 10919 Jumpstart fails to detect rules.ok
- 10952 Jumpstart does not mount all file systems
- 11972 Jumpstart boots interactive instead using rules.ok file
- 12195 Jumpstart: Install client boots from wrong server
- 11070 Jumpstart gets stuck in configuring /dev
- 12019 Jumpstart add\_install\_client Error:Unknown client "hostname"
- 12196 Jumpstart: install client doesn't boot, install server configured
- 12022 No network boot server, unable to install the system
- 12040 Requesting IP address
- 12171 WARNING: getfile: RPC failed: error 5 (RPC timed out)
- 6383 autoinstall does not use begin and finish script
- 11862 System hangs during boot net - install
- 10247 Autoinstall hands off timezone problems
- 13498 Jumpstart is going interactive, looking for timezone and NIS/NIS+ nameservice
- 16173 Jumpstart error Panic -boot: Could not mount filesystem
- 16278 Solaris Jumpstart client hangs after the "configuring devices..." line.  
(MultiHomed Boot Svr, InstallSvr Diff SubNet)
- 16678 Jumpstart reports Warning: Could not find matching rule in rules.ok

### Infodocs

- 12308 How to (re)install Jumpstart image on to disk
- 12063 Jumpstart model rules Platform Names and Groups
- 15834 How to disable Maint. Update (MU) from Jumpstart installation
- 15744 Setting up a network interface alias
- 23488 How to identify the Solaris version of a Jumpstart image
- 27188 Upgrading/Downgrading from one OS cluster to another
- 27794 Solaris 8 Jumpstart Guide
- 40134 Guidelines for setting up Jumpstart installations with DHCP boot

- 40820 Solaris Operating Environment software Jumpstart utilizing Flash Archive
- 44580 How to install Web Start Flash Archives on a Boot Environment using Live Upgrade (Command-Line)
- 49462 Solaris 8 Operating Environment software Jumpstart with a Boot Server only utilizing Flash Archive.
- 46472 Example of a multi-homed Jumpstart installation