Advanced Simulation Technology inc.
500 A Huntmar Park Drive
Herndon, Virginia 20170 U.S.A.
Tel. (703)471-2104 • Fax. (703)471-2108
www.asti-usa.com

ASTi
T4 ACE Diskless
Installation Guide

Document: DOC-01-TEL4-DSLS-1

Advanced Simulation Technology inc. 500A Huntmar Park Drive, Herndon, Virginia, 20170 USA
Revision B (July, 2010)

Product Name: T4 ACE Diskless

ASTi T4 ACE Diskless Installation Guide

ASTi

500-A Huntmar Park Drive

Herndon, VA 20170

Table of Contents

1.0. Overview

ACE diskless architecture provides the ability to create images using ACE Studio software. The images are either burned onto CDs or uploaded to the network boot server. The diskless Targets boot from either the CD or a server on the local network.

Diskless operation offers several advantages:

- Ends cold starting individual platforms, new software loads are incorporated into the boot image.
- Incorporates spares easily; simply update the DHCP file with the new mac addresses.
- Eliminates mechanical failure associated with continuous swapping of removable hard drive cartridges in order to provide different security levels.
- Add an additional level of security to your Target system by having it boot, load and run models fully embedded and without a hard drive.

2.0. CD Boot

The diskless Target loads the required software from the boot image on a CD. The CD is specific to each platform.

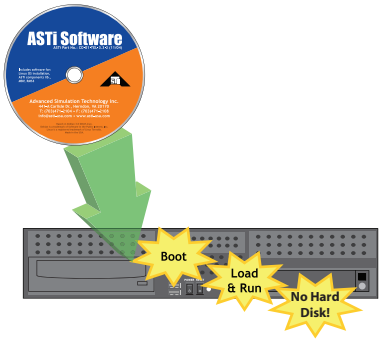


Figure 1: Local Boot

3.0. Network Boot

As an alternative to starting diskless Target systems from a CD, boot the Target with a boot image from a server over the local network.

3.1. The Server

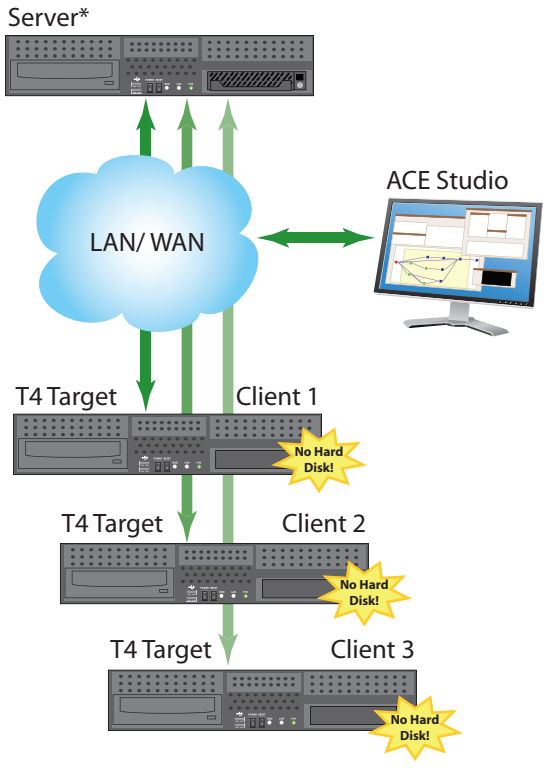
The server is either a standard Telestra 4 platform or a customer provided platform. The server contains two main components with regard to diskless operation. The first is a list of all of the valid diskless clients. Only clients that are on this list are able to load software remotely over the network from the server. Secondly, the server has a unique image for each client that is loaded and executed on each platform. The server has the ability to support unlimited diskless (network bootable) clients.

The server functionality is provided through the use of the following services:

- DHCP - This is needed for IP address assignment during PXE boot sequence.
- TFTP - The bootloader, kernel, and initial ram disk are downloaded via TFTP.

3.2. The Diskless Client

The client is a standard Telestra 4 platform but without a hard disk drive, as the name suggests. The client downloads the required software and ACE Project over the network into local memory, and then executes the model. Each client platform can have a unique embedded audio and communications model. For the end user, there is not a discernible difference between a disk-based system and a diskless system.



*Server may be provided by customer or ASTi.

Figure 2: Client Network Boot

3.3. Network Boot Sequence

When powered up, the client will issue a DHCP request onto the network. Via its DHCP daemon, the server receives this request and sends back a DHCP reply to the client. The reply contains the IP address of the client platform and other network settings. The client will then download the software components necessary to run the model.

The boot sequence over the network:

- Client sends out DHCP request.
- Server sends out DHCP response with IP address, TFTP server address, and bootloader file-name.
- Client downloads (TFTP) boot image and bootloader configuration file.

When the client is powered off all of the software packages (kernel, boot images, model, etc.) are erased from memory.

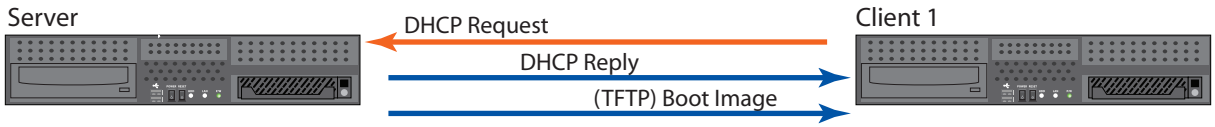


Figure 3: Client Network Bootup

4.0. Building Diskless Images

Whether you use a CD or the network boot to start your diskless Target, you first need to build the images using ACE Studio. This section details the procedure for building the images.

Note: Studio VM has a size limit of 15GB, if you are using Studio VM storage space for images is limited.

4.1. Package Mirrors

One of the first steps when building a diskless image is to create mirrors using DVD media provided by ASTi. These mirrors include one for the Base OS and one for the ACE software. Generally, these mirrors will provide all the software needed to build a working diskless image. In cases where additional software is needed that is not part of the standard ACE software, you may need to create a local mirror.

The three types of package mirrors are:

- Base - This is a complete operating system i.e. Red Hat® Enterprise Linux® or CentOS®. You must have a Base mirror in order to build an image.
- ACE - This is a complete ACE software release from ASTi, i.e. 4.16.
- Local - This mirror contains security updates or packages built by the customer. See Appendix A for more information on creating local mirrors.

The easiest way to create either an ACE or Base mirror is to use the ace-create-mirror script and the ASTi provided ACE and OS DVDs.

Note: Ensure that the correct DVD is in the system before running the corresponding script.

1. Insert the provided RHEL DVD into your system running ACE Studio, wait for file browser to pop-up.
2. Open a terminal shell by right-clicking on the desktop and selecting "Open Terminal."
3. Login as root using:

   ```
   su <enter>
   password: abcd1234
   ```

4. Run the following for the base software mirror:

   ```
   ace-create-mirror -n <name>
   ```

   e.g. `ace-create-mirror -n RHEL_5.3`

5. Eject RHEL DVD.
6. Insert ASTi ACE DVD.
7. Run the following for the ACE software mirror:

   ```
   ace-create-mirror -n ACE_<version>
   ```

   e.g. `ace-create-mirror -n ACE_4.16`

The package mirror setup may take some time, however each mirror is created once and can be reused among multiple images. The ace-create-mirror script will create the mirror files in the `/var/local/asti/mirrors/` directory which is used in section 4.3. Building Images.

4.2. Specifying Settings for Diskless Targets

The following sections describe how to configure and install the files to the `/var/diskless/` directory. These files are necessary for building diskless images.

4.2.1. Diskless Configuration Files

You will need to create a file that contains the network settings for the Targets you wish to operate as diskless Targets. This example configuration file combines the Comm Plan-style .ini syntax with the section headers from the system configuration file. Sections without the additional SECTION_NAME are treated as default values for all systems.

A generic section, as shown in the example below, applies the configuration parameters to all clients.

```
[network]
default-route = eth0
nameserver = 10.1.1.1
gateway = 10.2.0.254

[ntp]
server = 10.1.1.3
```

All sections that are IP address specific, as shown in the examples below, apply the configuration parameters to individual clients. Note the ‘f16-ffs’ is the name of the client in the first three examples below.

```
[eth0:f16-ffs]
mode = fixed
ip = 10.10.10.1
netmask = 255.255.0.0

[eth1:f16-ffs]
mode = fixed
ip = 192.168.49.219
netmask = 255.255.0.0

[eth2:f16-ffs]
mode = fixed
ip = 10.2.77.1
netmask = 255.255.0.0
```

```
[eth0:f16-ftd1]
mode = fixed
ip = 10.10.10.2
netmask = 255.255.255.0

[eth1:f16-ftd1]
mode = fixed
ip = 192.168.48.1
netmask = 255.255.0.0

[eth2:f16-ftd1]
mode = fixed
ip = 10.2.77.2
netmask = 255.255.0.0

[eth0:f16-ftd2]
mode = fixed
ip = 10.10.10.3
netmask = 255.255.255.0

[eth1:f16-ftd2]
mode = fixed
ip = 192.168.55.211
netmask = 255.255.0.0

[eth2:f16-ftd2]
mode = fixed
ip = 10.2.77.3
netmask = 255.255.0.0
```

To install the system configuration (.ini file) copy the xxxx.ini file to `/var/diskless/sysconfig/`. In the terminal type:

```
cp xxxx.ini /var/diskless/sysconfig/
```


4.2.2. Options Files

The options file is maintained separately from system backups. To install the options file (.tgz file) copy the file to /var/diskless/options/.

```
cp xxxx.tgz /var/diskless/options/
```

where *xxxx* is the name of the option file.

4.2.3. Backup Files

The application specific data for an image is loaded using the Backup/Restore feature in the Remote Management System. Backup files for diskless systems will typically contain the following items:

- Sound Library
- Project(s)
- Default Layout

To install the backup file (.tgz file) copy the file to /var/diskless/backups/.

```
cp xxxx.tgz /var/diskless/backups/
```

where *xxxx* is the name of the backup file.

Sound Libraries may be a separate backup file from the Project and Default Layout files. If you have a separate Sound Library backup file, install the backup file to the var/diskless/backups/ directory.

```
cp xxxxsoundlibrary.tgz /var/diskless/backups/
```

where *xxxx* is the name of the Sound Library backup file.

4.2.4. Custom Scripts

Contact ASTi for details on creating custom scripts. Use custom scripts to:

- Add host ssh keys to images for transferring files without a password
- Add security hardening scripts
- Add custom init. startup scripts

4.3. Building Images

To build the images, configure the file names and build a boot image using Live CD Creator to make an .iso image. Then either burn the .iso image to a CD or use the file to create a network boot image.

4.3.1. Configuring File Names

Before building the boot image, configure the image building configuration file names to match the previously created files (e.g. backup, options, etc.). Below is an example file.

Note: In the example below, 'RHEL 5.3' is the name of the base mirror and 'ACE_4.16' is the name of the ACE software mirror created in the **Package Mirrors** section of this document.

```
[main]
name=ACE Target LiveCD
description=Diskless image for ACE Target
label=AceTarget
type=target

# Options file to be installed in image

[options]
file=/var/diskless/options/F16_options_r0002.tgz

# Backup files to be installed in image (and restored)
[backup:SoundLib]
file=/var/diskless/backups/F16_sounds.tgz
description=Sound Library

[backup:Project]
file=/var/diskless/backups/F16_project.tgz
description=Project

# File that contains network settings for Targets that will use the image
[network]
file=/var/diskless/sysconfig/F16_network.ini

# Paths to mirrors used to create image
[base_mirror:rhel5]
url=file:///var/local/asti/mirrors/base/RHEL_5.3/
[ace_mirror:acetarget-4.16]
url=file:///var/local/asti/mirrors/ace/ACE_4.16/ace-rhel
```



```
[ace_mirror:astiextras]
url=file:///var/local/asti/mirrors/ace/ACE_4.16/astiextras/i386/

[ace_mirror:fedora-epel]
url=file:///var/local/asti/mirrors/ace/ACE_4.16/fedora-epel/5Client/i386/
```

4.3.2. Generating the ISO Image

Use the ace-create-image script and the Live CD Creator to build the boot image into an .iso image. In ACE Studio type the following to build a boot image:

```
ace-create-image xxxx.ini
```

where *xxxx* is the configuration .ini file.

This outputs target-livecd.ks file.

```
livecd-creator --config=target-livecd.ks --fslabel=yyyy
```

where *yyyy* is the name of the boot image.

The .iso image created by Live CD Creator is either used to burn a CD which is then used to boot the diskless Target or it is used to generate a network bootable image as outlined below.

4.3.3. Generating the Network Boot Image

Note: This section is for network boot only.

Pre-Execution Environment (PXE) boot images are network bootable images that are created from an existing .iso image, i.e. you must first run Live CD Creator as shown in section 4.3.2. In ACE Studio, use this command to generate PXE boot images:

```
livecd-iso-to-pxeboot
```

Note: This command must run as root.

Example:

```
livecd-iso-to-pxeboot Target-4.16.iso
```

where ‘Target-4.16.iso’ is the name of the .iso file created in section 4.3.2. Generating the ISO Image.

The script outputs the PXE boot image files into a directory called ‘tftpboot.’

Note: There is a known bug in network boot when images are 500MB or bigger the network boot will fail.

4.3.4. Updating the DHCP Server

Note: This section is for Network Boot only.

Follow the steps below to update the DHCP, where *xxx.xxx.xxx.xxx* is the IP address of the DHCP server.

1. Copy PXE boot image over to the DHCP Server and make sure to give the test directory a unique name.

   ```
   scp -r tftpboot root@xxx.xxx.xxx.xxx:tftpboot.target-4.16
   ssh root@xxx.xxx.xxx.xxx
   ```

2. Rsync the new ‘tftpboot’ directory to tftp area, make sure to include the trailing /.

   ```
   rsync -va --delete tftpboot.target-4.16/ /tftpboot/asti-boot/
   ```

This sets up the default network boot image for all systems using the server.

5.0. How to Configure DHCP/TFTP for Diskless

5.1. Overview

Target systems utilize the standard PXE to boot off a network server. Unlike previous ASTi implementations, Target systems can boot from any server that meets the following requirements. The services listed below are required on the network, but not necessarily on the same server.

- TFTP (Must support 'tsize' option).
- DHCP
- SSH/SCP

5.2. TFTP

The toolchain provided by ASTi for building diskless images produces a complete ‘tftpboot’ directory including the items below. The PXELINUX configuration directory contains a single file, ‘default’ containing the default boot settings. Since the same Target image can boot on multiple systems, most customers can simply use the default configuration. See below for file examples.

The ramdisk containing the Target file system.

```
initrd0.img
```

The PXE Linux boot loader:

```
pxelinux.0
```

The PXE Linux configuration directory:

```
pxelinux.cfg
```

The kernel image:

```
vmlinuz0
```

If more configurations are needed, please consult the PXELINUX documentation for instructions on how to match based on either MAC or IP address. PXELINUX documentation located at:

http://syslinux.zytor.com/wiki/index.php/PXELINUX

5.3. DHCP

The example below shows a minimal DHCP configuration file that supports booting a Target image. ASTi strongly recommends creating a separate group for Targets, especially since the DHCP server likely supports non-ASTi systems. The ‘filename’ option inside the ASTi group allows the use of the ‘tftpboot’ directory described above without modification.

A required command, which is specified globally or locally within the ASTi group, is `use-host-decl-names on;`. This command allows the system hostname assignment via the second DHCP request that occurs after the PXE boot sequence. In order to support reusable boot images, Target systems have the capability to configure system settings such as IP addresses at runtime based on the hostname provided by DHCP.

Using the hostname rather than MAC address to identify each Target also has the benefit of simplifying system replacement. Swapping in a spare Target requires only updating the MAC address contained in the DHCP configuration file.

```
# DHCP Server Configuration file.
#   see /usr/share/doc/dhcp*/dhcpd.conf.sample
deny unknown-clients;

allow booting;
allow bootp;

ddns-update-style ad-hoc;

option subnet-mask 255.255.255.0;
default-lease-time 86400;
max-lease-time 86400;

subnet 192.168.106.0 netmask 255.255.255.0 {
    }
```

```
group {
        # This option is required for proper Target system configuration.
         use-host-decl-names on;

        # PXE-specific options:
        next-server 192.168.106.1;
        filename "/ASTI_TELESTRA/pxelinux.0";
        host ace-target {
        hardware ethernet 00:16:76:38:c9:e6;
        fixed-address 192.168.106.100;
        }
}
```


Appendix A: Creating Local Mirrors

Introduction

For most diskless systems creating the base mirror and the ACE mirror will provide all the software needed to build a working diskless image. However, there are situations where a customer will need to create a local mirror using software packages not included in a standard ACE software release. These situations include:

- RHEL bug fixes
- RHEL security patches
- ASTi software updates
- Customer built RPM packages

The following example below shows how to create a local mirror for security updates and include it as part of a diskless image. The same basic steps can be used for other package types such as ASTi software updates or RHEL bug fixes.

During the image build, YUM determines which packages to install using standard RPM dependency and version information. This means you can specify multiple local mirrors in your configuration. For example, you can create a local mirror containing all security updates released over the past month. When the next round of security updates are released, you can simply add a new mirror and YUM will install the latest set of packages from both mirrors.

Example

Download Software Packages

Since ASTi is not authorized to distribute RHEL software updates, you will need download these directly from RHN or obtain the updates from your IA department. Once you have the updated packages, copy them to your Studio system.

Initialize Local Mirror Directory

The standard mirrors used to create a diskless image are located under `/var/local/asti/mirrors`. If you plan to use multiple local mirrors, ASTi recommends creating a directory structure to help organize the different types of mirrors.

Create local mirror structure, including mirror directory for March 2009 security updates.

```
mkdir -p /var/local/asti/mirrors/local/security/2009-03/
```

Copy RPM packages to mirror directory.

```
cp *.rpm /var/local/asti/mirrors/local/security/2009-03/
```

Create local mirror YUM metadata using createrepo utility.

```
createrepo /var/local/asti/mirrors/local/security/2009-03/
```

Add Local Mirror

At this point, your local mirror is initialized and ready to use in building a diskless image. The following lines will need to be added to your diskless image configuration file.

```
; March 2009 security updates
[local_mirror:2009-03]
url=file:///var/local/asti/mirrors/local/security/2009-03
```

After running the ace-create-image command, you can verify your configuration settings by looking for the following lines near the top of the target-livecd.ks file. There should be a section listing all local mirrors.

The Local Mirror Repositories:

```
repo --name=2009-03 --baseurl file:///var/local/asti/mirrors/local/
security/2009-03
```


Add Packages

If your local mirror contains only security updates, there are no extra steps required to include these updates when building a new image. The diskless image creation scripts will automatically select to install the latest version of a package.

If you wish to add packages that are not included in the default installation set you will need to add a new 'packages' section. This allows you to take advantage of the "%include" syntax supported by kickstart. This example adds the inotify-tools package to the diskless image. Start by initializing a local mirror that contains this package and add the mirror configuration to the INI file.

Since the target-livecd.ks file is generated by the ace-create-image script, create a new top-level kickstart file to manage adding additional kickstart files.

Based on the previous examples, here is the content of the new target-4.16.ks:

```
#Include file generated by ace-create-image
%include target-livecd.ks

#Include file for additional package installation
%include inotify-tools.ks
```

Here is the content of the inotify-tools.ks file:

```
#Additional packages not installed by default
%packages
inotify-tools
```

After creating these two new files, create the diskless image using the following command:

```
livecd-creator –config=target-4.16.ks –fslabel=yyyy
```

where *yyyy* is the name of the boot image.

Appendix B: Kickstart Installation Scripts

Creating diskless images based on the default configuration will produce a clean image similar to cold starting a disk-based Target. Just like a disk-based Target, customers will need to modify the system based on their specific requirements. Unlike a disk-based Target, diskless systems must be modified either during image creation or at startup, due to the lack of permanent storage.

This section contains several examples on how to customize a diskless image.

General Setup

Please read the previous section ‘Appendix A: Adding Packages’ for details on creating a top-level kickstart file. This appendix will show each example as individual kickstart files, which can be referenced from the top-level kickstart file.

The diskless tools utilize a chroot environment during the image creation process. In the chroot environment the root directory changes for a specific process, utilizing a separate virtualized copy of the operating system. Kickstart scripts can execute either inside or outside the chroot environment. When running a script inside the chroot, the root directory and available commands will operate similar to a Target system. When running a script outside, the chroot environment directories and commands will be based on the Studio/Host system. The special environment variable “$INSTALL_ROOT” is used outside the chroot to point to the start of the chroot file system.

SSH Tips and Tricks

SSH Server Keys

After a cold start, the ‘SSH’ server will generate a set of keys during the first boot. Unfortunately by default, diskless systems will generate these keys for **every** boot. This can be a problem for disk-based host systems that communicate via SSH. The kickstart example below shows how to generate the SSH server keys while creating the image so they will remain constant between diskless boots.

```
# Start/Stop the SSH server to generate the server keys
# This means they don't change between boots
%post
/etc/init.d/sshd start
/etc/init.d/sshd stop
%end
```


Add Public SSH Keys

Controlling a diskless system is primarily done using the SSH utilities. Downloading projects and soundfiles, restoring backups, and installing layouts are all examples of using SSH and SCP to modify a generic diskless image. By default, the SSH utilities require entering a pass phrase for initializing a connection. The examples below show how to install public/private keys for automating the SSH initialization when communicating to and from a Target.

```
## Adding Host's public SSH key.
%post --nochroot


SSH_KEY_FILE=id_rsa_f16.pub
SSH_KEY_PATH=/var/diskless/data/id_rsa.pub.f16
SSH_USER=root
SSH_HOME=$INSTALL_ROOT/$SSH_USER/.ssh
SSH_KEY_PERM=644
SSH_AUTH_PERM=600
SSH_AUTH_FILE=$SSH_HOME/authorized_keys

# Install key and create .ssh directory if needed.
mkdir -p $SSH_HOME
cp -vf $SSH_KEY_PATH $SSH_HOME/$SSH_KEY_FILE

# Set permissions on ssh key.
chmod $SSH_KEY_PERM $SSH_HOME/$SSH_KEY_FILE

# Setup key in authorized_keys file.
cat $SSH_HOME/$SSH_KEY_FILE > $SSH_AUTH_FILE

# Set permissions on authorized_keys file.
chmod $SSH_AUTH_PERM $SSH_AUTH_FILE

%end

# Install private key for diskless system.
%post --nochroot
SSH_KEY_FILE=id_rsa
SSH_KEY_PATH=/var/diskless/data/id_rsa_f16
SSH_USER=root
SSH_HOME=$INSTALL_ROOT/$SSH_USER/.ssh
```

```
SSH_KEY_PERM=600

cp -vf $SSH_KEY_PATH $SSH_HOME/$SSH_KEY_FILE

# Set permissions on ssh key.
chmod $SSH_KEY_PERM $SSH_HOME/$SSH_KEY_FILE
%end
```

Modify the "/etc/hosts" File

```
# Install custom hosts file.
%post --nochroot

HOST_FILE=/var/diskless/data/host-f16
CUSTOMER_HOST_FILE=$INSTALL_ROOT/etc/hosts.customer

cp -vf $HOST_FILE $CUSTOMER_HOST_FILE

%end
```

Add an Initialization Script

Initialization scripts run at boot time and can be used for customizing a system at runtime. The file `/usr/share/doc/initscripts-8.45.30/sysvinitfiles` contains details on writing initialization scripts. Please note this example requires running scripts inside and outside of the chroot environment.

```
# Install startup script
%post --nochroot

INIT_SCRIPT=/var/diskless/data/my_startup_script
INIT_DIR=$INSTALL_ROOT/etc/init.d/

cp -vf $INIT_SCRIPT $INIT_DIR
chmod 755 $INIT_DIR/my_startup_script

%end

%post
/sbin/chkconfig my_startup_script reset
%end
```


Change the Root Password

The example below shows clear text passwords. The “chpasswd” utility also supports MD5 and DES encrypted passwords.

```
%post
echo “root:abcd5678” | chpasswd
%end
```

Appendix C: Installing Telestra 4 Security Software Packages

This section shows how to incorporate ASTi's Security Software Package into a diskless image build. The first step is to create a local mirror which includes the security software RPM. Appendix A provides instructions on how to create a local mirror and add the mirror to your diskless image configuration file.

After adding the local mirror, you will need to create a kickstart installation script to add the RPM and execute the security installation script. The following kickstart script example shows how to add and install the 1.8-1 version of security software.

Refer to section 6.0 of the Telestra 4 Target Cold Start Procedure (DOC-02-TEL4-TCS-1) for the latest instructions on installing the Security Software Package.

```
# Include the ASTi Security Software RPM
%packages
asti-rhel5-security

# Execute the Security script
%post
/usr/local/bin/secure_telestra.sh
%end
```


Appendix D: Initialization Script Priority Road Map

The ACE Target software boots into run level 4 by default. Within this run level, initialization scripts start based on a priority level. Scripts using the same priority level are started alphabetically.

This section lists the priority level for several important initialization scripts. This information can help you determine when to start your own custom initialization scripts.

The following command will show all startup scripts for run level 4.

```
ls -1 /etc/rc4.d/S*
```

S10network

This script is the beginning of the diskless network initialization. Before this script runs, no network access is available. After this script runs, the `eth0` network interface is initialized using DHCP.

S11diskless_network

This script translates the diskless_network.ini global network settings into the specific settings for the individual Target. After this script executes, all configured network interfaces are available.

S12syslog

Any script that runs prior to S12 will not have messages logged through syslog.

S13iscsi

If you are using iSCSI services, you will need to configure iSCSI settings before this script runs.

S55sshd

After this service starts, you will be able to remotely login or use SSH services such as SFTP or SCP. Note: SSH commands can run on the Target itself after the S11diskless_network script executes.

S81

This priority level is a general starting point for ACE software daemons. Configuration items such as options files should be available before this priority level.

S99

This is typically the last priority level used during startup. One script that executes at this priority is S99layout. This script will install the default layout for the ACE Target. Any dependencies such as the project, sound files, and default layout selection must be in place before this priority level.

