



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

Telestra Diskless Installation Guide

Document: DOC-TEL-DSLS-IG-C-1

Product Name: ACE Target and ACE Studio

Telestra Diskless Installation Guide

© Copyright ASTi 2018

Restricted Rights: Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

This material may be reproduced by or for the U.S. Government pursuant to the copyright license under the clause at DFARS 252.227-7013 (1994).

ASTi

500A Huntmar Park Drive

Herndon, Virginia 20170 USA

Revision history

Date	Revision	Version	Comments
09/25/2017	C	0	Edited content for clarity and accuracy; updated content to include Red Hat 6 instructions.
07/10/2018	C	1	Updated ACE-RHN reference to ACE-RHSM in "Download software packages."

Contents

1.0 Introduction	1
1.1 Required equipment	2
2.0 Create package mirrors	3
3.0 Define diskless Telestra Target settings	4
3.1 Create Telestra Target network settings file	4
3.2 Install Options files	5
3.3 Install backup files	5
4.0 (Optional) Install Telestra security packages	8
5.0 Build diskless images	9
5.1 Create image configuration file	9
5.2 Generate ISO image	11
6.0 (Optional) Configure boot server for diskless operation	12
6.1 Convert network boot image into PXE	13
6.2 Install images on TFTP boot server	14
6.3 Edit DHCP configuration file	15
7.0 Telestra diskless installation checklist	17
Appendix A: Create local mirrors	18
A-1 Download software packages	18
A-2 Initialize local mirror directory	18
A-3 Add local mirror	19
A-4 Install additional packages	19
Appendix B: Kickstart installation scripts	21
B-1 Generate SSH server keys	21
B-2 Add public SSH keys	22
B-3 Add initialization script	23
B-4 Change root password	24

B-5 Install HLA/RTI script	24
B-6 Install TTS software, libraries, and voice data	26
Appendix C: Initialization script priority road map	28

1.0 Introduction

The Telestra diskless boot feature enables the Telestra Target to boot into a fully functional configuration without an internal hard drive. With ACE diskless build tools, create bootable images on an ACE Studio development workstation, and use these images to boot a diskless Target in one of two ways:

- Burn the image to a portable storage device (i.e., CD, DVD), and insert it into the Telestra server, which can boot from that device.
- Copy the image to a network boot server on the local network. Configure the Target and boot server so the Target can boot over the network using Preboot Execution Environment (PXE).

Diskless operation offers several advantages:

- Update or cold start one boot server rather than multiple, individual platforms. The boot image incorporates software loads; as a result, connected Targets automatically update when you update the boot server and reboot the Target.
- Easily incorporate spares; simply update the Dynamic Host Configuration Protocol (DHCP) file with the new MAC addresses.
- Boot, load, and run fully embedded models without a hard drive, increasing your facility's security.
- Experience fewer mechanical failures resulting from continuous hard drive removal.

The *Telestra Diskless Installation Guide* describes how to configure, build, and deploy bootable images using an ACE Studio development workstation. It also explains how to configure the Target to boot from a CD or DVD drive or over the network using PXE. Additionally, this guide describes how to configure a network boot server and the required DHCP server, customize diskless builds using custom Kickstart scripts, and install ASTi security software packages on a diskless Target.

This document is intended for maintenance technicians or network engineers who are familiar with Linux terminology, IP networking concepts, and PXE network boot concepts, if applicable. Before you begin the Telestra ACE Diskless Installation Procedure, plan out your facility's network settings with host names, IP addresses, etc. Additionally, ensure the latest version of ACE Studio is installed on your development workstation. To check the latest ASTi software releases, go to the [ACE Software Release Notes](#). If needed, complete the ACE Studio cold start procedure found in the [ACE Target and Studio Cold Start Guide](#) or [ACE Studio VM Quick Start Guide](#).

1.1 Required equipment

The following equipment is required to install a diskless Telestra Target:

- ACE Studio development workstation
- Red Hat 5 Installation DVD or Red Hat 6 Installation DVD
- Telestra Target Software Installation DVD
- Telestra Target running ACE Target software
- *(Optional)* Portable storage device (e.g., CD, USB drive)
- *(Optional)* Boot server (e.g., Telestra Target or customer-provided server)

2.0 Create package mirrors

To build diskless images, first create mirrors using ASTi's provided media. During this step, you will create mirrors for both the base operating system and the ACE software. These mirrors provide all the software needed to build a functional diskless image. If you need additional software not included as part of the standard ACE package, you may need to create a local mirror.

Three types of package mirrors exist:

- *Base mirror*: contains 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 mirror*: contains a complete ACE software release from ASTi (e.g., version 6.7.0).
- *Local mirror*: contains any security updates or packages that you build. To create local mirrors, go to A-3, "Add local mirror" on page 19.

To create base and ACE package mirrors, follow these steps:

1. Turn on the ACE Studio development workstation, and wait for it to boot.
2. Log into ACE Studio using the following default credentials:

Username	Password
aceuser	aceuser



Note: If your organization changed these default credentials, check with your IT administrator for an updated username and password.

3. Insert the Red Hat 5 Installation DVD or Red Hat 6 Installation DVD into the media drive. Wait for the file browser to open.
4. To open a new terminal, right-click the desktop, and select **Open in Terminal**.
5. To log into the system as root, at the prompt, run the **su** command, and enter **abcd1234** for the password.
6. To create a base software mirror, run **ace-create-mirror -n RHEL_X.X**, where *X.X* represents your Red Hat system's release identifier (e.g., 6.8).
7. When the process is complete, eject the Red Hat Installation DVD.
8. Insert the Telestra Target Software Installation DVD into the drive.
9. At the prompt, run **ace-create-mirror -n ACE_X.X.X**, where *X.X.X* represents ACE software's update release identifier (e.g., 6.7.0).

Package mirror setup may take a few minutes; however, you only need to create each mirror once, and you can reuse mirrors among multiple images. The **ace-create-mirror** command creates the mirror files in the **/var/local/asti/mirrors** directory.

3.0 Define diskless Telestra Target settings

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

3.1 Create Telestra Target network settings file

First, create a system configuration file in initialization (i.e., `.ini`) file format. This file contains network settings for Telestra servers that will operate as diskless Targets. As a result, you can customize settings for each system at boot time. To create this network settings file, follow these steps:

1. On an ACE Studio development workstation, at the terminal prompt, log into the system using the following default credentials:

Username	Password
root	abcd1234

2. Go to the `/var/diskless` directory.
3. Using the text editor of your choice, create a new file named `settings.ini`, where `settings` represents the name of the Target network settings file.
4. Within the file, define the configuration parameters for each Target connected to the boot server. The following examples combine Comm Plan-style `.ini` syntax with the section headers from the system configuration file. Sections that do not specify a bracketed section name represent default values for all systems.

To apply generic configuration parameters to all Target systems, use the following syntax:

```
[network]
gateway-device = eth0
nameserver = xxx.xxx.xxx.xxx
gateway = yyy.yyy.yyy.yyy
hostname = localhost

[ntp]
server = zzz.zzz.zzz.zzz
```

where `xxx.xxx.xxx.xxx` represents the network's nameserver address, `yyy.yyy.yyy.yyy` represents the gateway address, and `zzz.zzz.zzz.zzz` represents the Network Time Protocol (NTP) server's address.

5. If you are booting multiple Target systems on a shared image, apply the following configuration parameters for each one:

```
[ethX: hostname]
mode = dhcp
ip = xxx.xxx.xxx.xxx
netmask = yyy.yyy.yyy.yyy
```

In the example above, *X* represents the Ethernet port number (e.g., eth0, eth1), and *hostname* represents the name of the Target host. Additionally, *xxx.xxx.xxx.xxx* represents the port's IP address, whereas *yyy.yyy.yyy.yyy* represents the netmask address.

6. Save and close the file.
7. To install the file, at the prompt, run **cp settings ini /var/diskless/sysconfig**.

3.2 Install Options files

The Options file is an archive (.tgz) file maintained separately from system backups. To install the Options file, follow these steps:

1. Find the Options file on the Telestra Options File CD or on the Telestra Target. If the Options file is located on the Target, go to **/user/local/ASTi/options**, and find the .tgz file.
2. Using a CD or USB drive, transfer the Options file to the ACE Studio development workstation.
3. On the development workstation, at the terminal prompt, run **cp project-XXX_R00N.tgz /var/diskless/options**, where *project-XXX* represents your project's abbreviated name and three-digit number, and *N* represents the Options file's release number.

3.3 Install backup files

The application-specific data for an image is loaded using the **Backup/Restore** page in the Remote Management System (RMS). Backup files for diskless systems typically contain the following items:

- Sound library
- Project(s)
- Default layout

For more information about backing up your system, go to "Backup System Configuration" in the [Telestra Remote Management System User Guide](#).

To install backup files, follow these steps:

1. Open a web browser on a computer sharing a network with the Telestra server.
2. To access the RMS, in the address bar, enter the Telestra server's IP address.
3. In the top-right corner, select **Login**.
4. Log into the system using the following default credentials:

Username	Password
admin	astirules

5. From the left menu, under **Configuration**, go to **Backup Restore**.
6. On the **System Backup/Restore** page, go to **Backup System Configuration**.



Figure 1: Backup System Configuration

7. Select all or some of the configuration options, and select **Start Backup**.

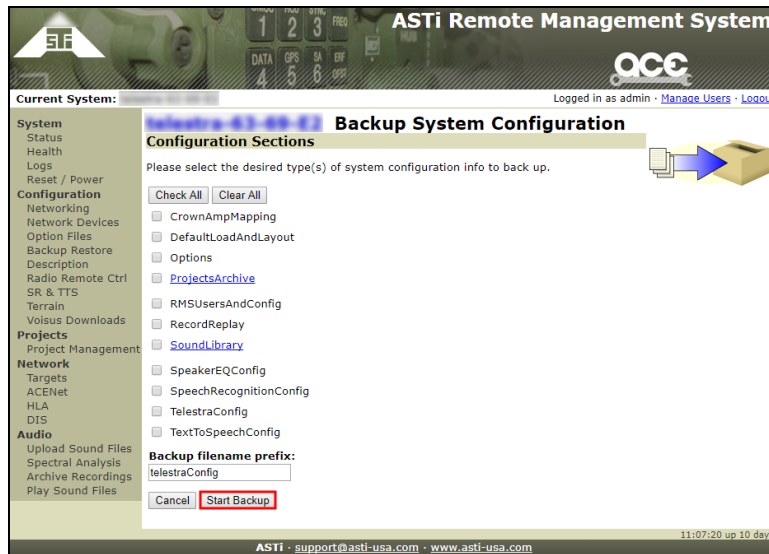


Figure 2: Start Backup

8. When the backup is complete, next to **Filename**, select the archive (.tgz) file, and the backup downloads your local system.
9. Open a terminal, and find the backup file on your local system.
10. At the prompt, run **cp backup_file.tgz /var/diskless/backups**, where *backup_file* represents the backup file's name.
11. (Optional) If you have a separate backup file for sound libraries, run **cp filesound-library.tgz /var/diskless/backups**.

4.0 (Optional) Install Telestra security packages

This section explains how to incorporate ASTi's security software package into a diskless image build. To install Telestra security packages, follow these steps:

1. On the ACE Studio development workstation, create a local mirror that includes the security software Red Hat Package Manager (RPM). To create and add a local mirror to your diskless image configuration file, go to Appendix A, "Create local mirrors" on page 18.
2. To include the ASTi security software RPM, add the following:

```
%packages
asti-rhel5-security
```

3. To create and execute a Kickstart installation script, go to Appendix B, "Kickstart installation scripts" on page 21. The following Kickstart script example shows how to add and install the 1.8-1 version of security software:

```
%post
/usr/local/bin/secure_telestra.sh
%end
```

4. To install the latest security package, go to the [IA Software Installation Guide](#).

5.0 Build diskless images

To build an International Standards Organization (ISO) image, configure the file names and build a boot image with **Live CD Creator**. Then burn the ISO image to a CD, or use the file to create a network boot image.



Note: The ACE Studio virtual machine (VM) has a size limit of 15 GB; image storage space is limited.

5.1 Create image configuration file

Before building the boot image, configure the image-building configuration file names to match the previously created files (e.g., backup, options). To configure these file names, follow these steps:

1. On the ACE Studio development workstation, go to a working directory in the root directory.
2. Using the text editor of your choice, create a new file named *imgconfig.ini*, where *imgconfig* represents the name of the image configuration file. Create as many configuration files as needed for your network setup.
3. Within the file, enter generic network information for all Telestra Targets. This information includes the diskless server's name, description, label, and type, as shown in the example below:

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

4. Define the path and name of Options file(s) included in the image:

```
[options]
file=/var/diskless/options/project-XXX_r000N.tgz
```

where *project-XXX* represents your project's abbreviated name and three-digit number, and *N* represents the Options file's release number.

5. Define the path, name, and description of the sound library and project backup files included in the image:

```
[backup:SoundLib]
file=/var/diskless/backups/file_name.tgz
description=Sound Library

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

where *file_name* represents the sound library or project backup file's name.

6. Define the path and name of the *settings.ini* file that you created in Section 3.1, "Create Telestra Target network settings file" on page 4.

```
[network]
file=/var/diskless/sysconfig/settings.ini
```

7. Define the path to the base mirror you created in Section 2.0, "Create package mirrors" on page 3.

```
[base_mirror:rhelN]
url=file:///var/local/asti/mirrors/base/RHEL_X.X/
```

In the example above, *N* represents your Red Hat operating system's major release identifier (e.g., rhel6), while *X.X* represents its minor release identifier (e.g., RHEL_6.8).

8. Define the path to the ACE mirror you created in Section 2.0, "Create package mirrors" on page 3.
 - a. For Red Hat 5 systems, use the following syntax:

```
[ace_mirror:acetarget-X.XX]
url=file:///var/local/asti/mirrors/ace/ACE_X.XX/ace-rhel
```

where *X.XX* represents your ACE software's minor release identifier (e.g., ACE_4.50).

- b. For Red Hat 6 systems, use the following syntax:

```
[ace_mirror:acetarget-X.X.X]
url=file:///var/local/asti/mirrors/ace/ACE_X.X.X
```

where *X.X.X* represents your ACE software's update identifier (e.g., ACE_6.7.0).

9. For Red Hat 5 or lower systems, define the file paths to additional packages required for the build. These scripts account for directory structure differences between Red Hat 5 and 6.

```
[ace_mirror:astiextras]
url=file:///var/local/asti/mirrors/ace/ACE_X.XX/astix-
tras/i386/

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

10. Save your changes, and close the file.

5.2 Generate ISO image

To build the boot image into an International Standards Organization (ISO) image using **Live CD Creator**, follow these steps:

1. On the ACE Studio development workstation, at the prompt, run **ace-create-image settings.ini**, where *settings.ini* represents the Telestra Target network settings initialization file you created in Section 5.1, "Create image configuration file" on page 9.
2. To create the **target-livcd.ks** file, run one of the following commands:
 - a. *Red Hat 5*: **livecd-creator --config=target-livcd.ks --fslabel= telestra-X.XX.iso**, where *X.XX* represents the operating system's minor release identifier.
 - b. *Red Hat 6*: **livecd-creator --config=target-livcd.ks --fslabel= telestra-X.X.X.iso**, where *X.X.X* represents the operating system's update release identifier.
3. Wait for **Live CD Creator** to generate the image. This process could take anywhere from 10–30 minutes, depending on your system's operating speed. When the command prompt reappears, image generation is complete.
4. Use the ISO image to burn a CD, which you will then use to boot the diskless Target or generate a Preboot Execution Environment (PXE) network boot image. For more information, go to Section 6.1, "Convert network boot image into PXE" on page 13.

6.0 (Optional) Configure boot server for diskless operation

Instead of starting diskless Telestra Target systems from a CD, you can boot the Target with a boot image from a server over the local network. The boot server is either a standard Telestra platform or a customer-provided platform. The server contains two main components:

- A list of all the valid diskless clients; only clients on this list can load software remotely over the network from the boot server.
- A unique image for each client that is loaded and executed on each platform; the server can support unlimited diskless clients.

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

- *Dynamic Host Configuration Protocol (DHCP)*: needed for IP address assignment during the Pre-Execution Environment (PXE) boot sequence.
- *Trivial File Transfer Protocol (TFTP)*: used to download the bootloader, kernel, and initial RAMDisk.

The client is a standard Telestra 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. The end user will not notice a discernible difference between a disk-based and a diskless system.

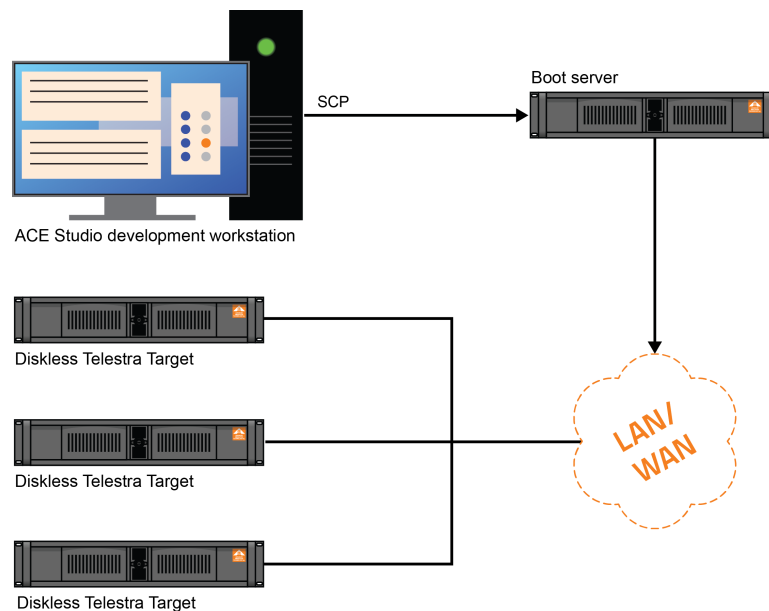


Figure 3: Diskless Telestra network diagram

When powered up, the client issues a DHCP request on 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 then downloads the software components necessary to run the model.

The network boot sequence is as follows:

1. The client sends out a DHCP request.
2. The boot server sends a DHCP response with its IP address, TFTP server address, and bootloader file name.
3. The client downloads the TFTP boot image and bootloader configuration file.

When the client is powered off, all of the software packages (i.e., kernel, boot images, model, etc.) erase from its memory.

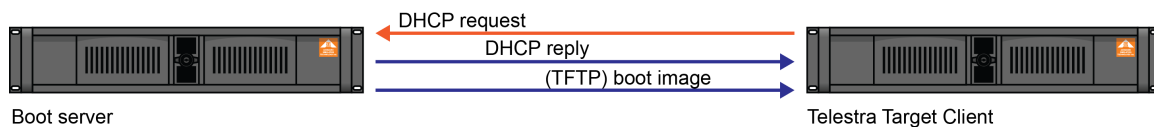


Figure 4: Client network start-up

6.1 Convert network boot image into PXE

Pre-Execution Environment (PXE) is a method that allows the client to boot with only its network card. As long as the client is connected to the network and supports this standard, you can circumvent the normal boot procedure. These boot images are derived from the existing International Standards Organization (ISO) image you created in Section 5.2, "Generate ISO image" on page 11.

To convert the network boot image into PXE, follow these steps:

1. On the boot server, at the prompt, run **livecd-iso-to-pxeboot *image_name*.iso**, where *image_name* represents the name of the ISO image (e.g., Target-4.50.iso or Target-6.2.0.iso).



Caution: When images are 500 MB or bigger, the network boot fails.

- The script outputs the PXE boot image files into the Trivial File Transfer Protocol (TFTP) boot directory. By default, TFTP boot is located in `/var/lib/tftpboot`, although its location may vary depending on your system's configuration. Go to your system's TFTP boot directory, and confirm that the following files exist:

Name	Description
initrd0.img	The RAMDisk containing the Telestra Target file system.
pxelinux.0	The PXE Linux bootloader.
pxelinux.cfg	The PXE Linux configuration directory, which is relative to the prelinux.0 boot loader. This directory contains a single default file containing the boot settings. Since the Target image can boot on multiple systems, most customers can simply use the default configuration.
vmlinuz0	The kernel image.

Table 1: TFTP boot directory contents



Note: To match based on your MAC or internet protocol IP address, go to [PXELINUX](#).

6.2 Install images on TFTP boot server

Next, install network boot images on the Trivial File Transfer Protocol (TFTP) boot server, and set the default network boot image for all connected systems. To install the International Standards Organization (ISO) and network boot images to the TFTP boot server, do one of the following:

- Copy the Pre-Execution Environment (PXE) boot image to the TFTP boot server via Secure Copy (SCP). At the prompt, run **scp -r tftpboot root@xxx.xxx.xxx.xxx:tftpboot.image_name**.

In the above command, `xxx.xxx.xxx.xxx` represents the TFTP boot server's IP address, and `image_name` represents the image you created in Section 6.1, "Convert network boot image into PXE" on the previous page.

- Copy the TFTP boot directory to the TFTP area via rsync. At the prompt, run **rsync -va -delete tftpbootimage_name.isol /tftpboot/asti-boot/**.

6.3 Edit DHCP configuration file

Next, configure the Dynamic Host Configuration Protocol (DHCP) server to support booting a Telestra Target image. The following instructions detail basic settings that your DHCP file should include to support diskless booting. Additionally, these instructions assume that your DHCP server is running a Linux-based operating system. Configuration steps may vary for Windows platforms.

To edit your network's DHCP configuration file, follow these steps:

1. On the DHCP boot server, install DHCP daemon software per your operating system's instructions. For more information about DHCP Linux installation, go to [Configuring a DHCPv4 Server](#).
2. Go to your DHCP configuration file. On Linux platforms, settings are typically stored in the `etc/dhcp.conf` directory.
3. To prevent unknown clients from receiving IP assignments, add the following using the text editor of your choice:

```
deny unknown-clients;
```

4. To enable diskless booting, add the following:

```
allow booting;  
allow bootp;
```

5. To implement the ad-hoc Domain Name System (DNS) update mode, add the following:

```
ddns-update-style ad-hoc;
```

6. To configure the DHCP server's lease range parameters, add the following:

```
option subnet-mask xxx.xxx.xxx.xxx  
default-lease -time XXXXX;  
max-lease-time YYYYY;
```

where `xxx.xxx.xxx.xxx` represents the subnet mask specific to the lease range, `XXXXX` represents the default IP lease time in seconds, and `YYYYY` represents the maximum IP lease time.

7. To set the DHCP server's subnet mask and netmask, add the following:

```
subnet xxx.xxx.xxx.xxx netmask yyy.yyy.yyy.yyy {  
}
```

8. Because your DHCP server most likely supports non-ASTi systems, create a separate group for Targets. Within this group, configure the following:
 - a. The **use-host-decl-names on;** directive, which allows the system hostname assignment via the second DHCP request that occurs after the Preboot Execution Environment (PXE) boot sequence. To support reusable boot images, Target systems can configure system settings such as IP addresses at run-time based on the hostname that the DHCP server provides.
 - b. PXE-specific options, including the following:
 - *next-server*: specifies the host address of the server that loads the initial boot file (i.e., *xxx.xxx.xxx.xxx*).
 - *filename*: allows use of the Trivial File Transfer Protocol (TFTP) boot directory described in Section 6.1, "Convert network boot image into PXE" on page 13. The filename path is relative to the TFTP directory's root and depends on your network's directory structure. In the example below, the filename path is **/ASTi_Telestra/pxelinux.0**.
 - *hardware ethernet*: specifies the Target's Ethernet address (i.e., *XX:XX:XX:XX:XX:XX*).
 - *fixed address*: specifies the fixed IP address assigned to the client (i.e., *yyy.yyy.yyy.yyy*).

```
group {  
  use-host-decl-names on;  
  
  #PXE-specific options  
  next-server xxx.xxx.xxx.xxx;  
  filename "/ASTi_Telestra/pxelinux.0";  
  host ace-target {  
    hardware ethernet XX:XX:XX:XX:XX:XX;  
    fixed-address yyy.yyy.yyy.yyy;  
  }  
}
```

9. To install the updated image, reboot all diskless Targets connected to the boot server.

7.0 Telestra diskless installation checklist

To perform the Telestra ACE Diskless Installation Procedure, you should have completed the following tasks:

- Create package mirrors.**
- Define diskless Telestra Target settings:**
 - Create Telestra Target network settings file.
 - Install Options files.
 - Install backup files.
- (Optional)* **Install Telestra security packages.**
- Build diskless images:**
 - Create image configuration file.
 - Generate International Standards Organization (ISO) image.
- (Optional)* **Configure boot server for diskless operation:**
 - Convert network boot image into Preboot Execution Environment (PXE).
 - Install images on Trivial File Transfer Protocol (TFTP) boot server.
 - Edit Dynamic Host Configuration Protocol (DHCP) configuration file.

Appendix A: Create local mirrors

For most diskless systems, creating the base mirror and the ACE mirror provides all the software needed to build a working diskless image. In some situations, a customer needs to create a local mirror using software packages not included in a standard ACE software release. These situations include the following:

- Red Hat bug fixes
- Red Hat security patches
- Customer-built Red Hat Package Manager (RPM) packages

The following sections show 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 Red Hat bug fixes.

During the image build, Yellowdog Updater, Modified (YUM) determines which packages to install using standard RPM dependency and version information. As a result, you can define 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, add a new mirror, and YUM installs the latest set of packages from both mirrors.

A-1 Download software packages

You must download software packages directly from Red Hat Subscription Management (RHSM) or obtain the updates from your Information Assurance (IA) department. Once you have the updated packages, copy them to your ACE Studio system.

A-2 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. To initialize a local mirror directory, follow these steps:

1. On the ACE Studio development workstation, create a local mirror structure, including a mirror directory for the most recent security update. At the prompt, run the `mkdir -p /var/local/asti/mirrors/local/security/YYYY-MM/` command, where `YYYY-MM` represents the year and month of the update's release (e.g., 2018-07 for July 2018).
2. Copy Red Hat Package Manager (RPM) packages to the mirror directory. At the prompt, run the `cp *.rpm /var/local/asti/mirrors/local/security/YYYY-MM/` command.
3. Create local mirror Yellowdog Updater, Modified (YUM) metadata using the `createrepo` utility. At the prompt, run the `createrepo /var/local/asti/mirrors/local/security/YYYY-MM/` command.

A-3 Add local mirror

At this point, your local mirror is initialized and ready to use during the build. On the ACE Studio development workstation, add the following security update lines to your diskless image configuration file:

```
[local_mirror: YYYY-MM]
url=file:///var/local/asti/mirrors/local/security/YYYY-MM
```

where *YYYY-MM* represents the year and month of the most recent security update (e.g., 2018-07 for July 2018).

After running the **ace-create-image** command, verify your configuration settings by looking for the following lines near the top of the **target-livecd.ks** file. You should see a section listing all local mirror repositories.

```
repo --name=YYYY-MM --baseurl file:///var/local/asti/mirrors/local/
security/YYYY-MM
```

A-4 Install additional packages

If your local mirror contains only security updates, no extra steps are required to build a new image. The creation scripts automatically install the latest version of a package. To add packages that are not included in the default installation, you must install a package that monitors file system events (i.e., creating and deleting files). To install this package, follow these steps:

1. On the ACE Studio development workstation, Create a new **target-X.X.X.ks** file, where *X.X.X* represents the latest Telestra version (e.g., 6.7.0). This file will manage adding Kickstart files.
2. In the **target-X.X.X.ks** file, enter the following:
 - The **target-livecd.ks** file generated using the **ace-create-image** script.
 - The **inotify-tools.ks** file for additional Kickstart package selection. This file will contain Kickstart commands to install the **inotify-tools** package.

```
%include target-livecd.ks
%include inotify-tools.ks
```

3. In the **inotify-tools.ks** file, create a new **packages** section that adds the **inotify-tools** Red Hat Package Manager (RPM) to the diskless build:

```
%packages
inotify-tools
```

4. To create the diskless image, at the prompt, run **livecd-creator --config=target-X.X.X.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 produces a clean image similar to cold-starting a disk-based Telestra Target. Just like a disk-based Target, you must modify the system based on your specific requirements. However, you must modify diskless systems during image creation or at start-up due to the lack of permanent storage. This section contains several examples on how to customize a diskless image.

To create a top-level Kickstart file, go to A-4, "Install additional packages" on page 19. This appendix shows each example as individual Kickstart files that you can reference from the top-level Kickstart file.

The diskless tools use a chroot environment during the image creation process. In the chroot environment, the root directory changes for a specific process, using 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 environment, the root directory and available commands operate similarly to a Target system. When running a script outside, the chroot environment directories and commands are based on the ACE Studio or host system. The special environment variable `$INSTALL_ROOT` is used outside the chroot environment to point to the start of the chroot file system.

B-1 Generate SSH server keys

Once you complete the cold start procedure, the Secure Shell (SSH) boot server generates a set of keys during the first boot. By default, diskless systems generate these keys for every boot. This behavior can be a problem for disk-based host systems that communicate via SSH.

To generate the server keys, start and stop the SSH boot server. The Kickstart example below shows how to generate the SSH server keys while creating the image so they remain constant between diskless boots:

```
%post
/etc/init.d/sshd start
/etc/init.d/sshd stop
%end
```

B-2 Add public SSH keys

You can control a diskless system using the Secure Shell (SSH) utilities, thus automating the login process. You might use SSH and Secure Copy (SCP) to modify a generic diskless image in one of the following ways:

- Download projects and sound files
- Restore backups
- Install layouts

By default, the SSH utilities require entering a pass phrase that initializes a connection with a Telestra Target. To automate SSH initialization, you must install a public SSH or private SSH keys.

To install public SSH keys, follow these steps:

1. On the ACE Studio development workstation, in the Kickstart file, define the variables for your specific application, as shown in the example below:

```
%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
```

2. Install the key, and create an **.ssh** directory, if needed:

```
mkdir -p $SSH_HOME
cp -vf $SSH_KEY_PATH $SSH_HOME/$SSH_KEY_FILE
```

3. Set permissions on the SSH key:

```
chmod $SSH_KEY_PERM $SSH_HOME/$SSH_KEY_FILE
```

4. Set up the key in an **authorized_keys** file:

```
cat $SSH_HOME/$SSH_KEY_FILE > $SSH_AUTH_FILE
```

5. Set permissions on the **authorized_keys** file:

```
chmod $SSH_AUTH_PERM $SSH_AUTH_FILE  
%end
```

6. Install a private key for diskless systems:

```
%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
```

7. Set permissions on the SSH key:

```
chmod $SSH_KEY_PERM $SSH_HOME/$SSH_KEY_FILE  
%end
```

B-3 Add initialization script

Initialization scripts run at boot time and can be used for customizing a system at run-time. The `/usr/share/doc/initscripts-8.45.30/sysvinitfiles` file contains details on writing initialization scripts.



Note: This example requires running scripts inside and outside the chroot environment.

The example below shows how to install a start-up script:

```
%post --nochroot
INIT_SCRIPT=/var/diskless/data/start-up_script
INIT_DIR=$INSTALL_ROOT/etc/init.d/
cp -vf $INIT_SCRIPT $INIT_DIR
chmod 755 $INIT_DIR/start-up_script
%end

%post
/sbin/chkconfig start-up_script reset
%end
```

where *start-up script* represents the script's name.

B-4 Change root password

The example below shows clear text passwords. The **chpasswd** command also supports MD5 and Data Encryption Standard (DES)-encrypted passwords.

```
%post
echo "root:abcd5678" | chpasswd
%end
```

B-5 Install HLA/RTI script

To install and configure a high-level architecture (HLA) run-time infrastructure (RTI), follow these steps:

1. ASTi does not provide RTI archives. On the ACE Studio development workstation, download the RTI archive from your provider's website.
2. Copy the file to the **/var/diskless/data** directory.

3. Create a new file for the **hla.ks** script, and force-copy the RTI archive to the chroot environment:

```
%post --nochroot
cp -vf /var/diskless/data/rti_file.tar.gz $INSTALL_ROOT/tmp
%end
```

where *rti_file.tar.gz* represents the RTI archive (e.g., **rtis_D33A_linux_g++-4.4.tar.gz** for run-time infrastructure (simulation), version D33A).

4. In the **hla.ks** file, include the following:
 - a. Make a parent-level install directory. The following example uses the **mkdir** command to create a parent directory named **/opt/rti**:

```
%post
mkdir -p /opt/rti
```

- b. Unpack the RTI archive, extracting the files to the directory you made in Step 4a. The following example uses the **tar xf** command to extract the **/tmp/rti_file.tar.gz** file. It then changes to the **/opt/rti** directory with the **-C** command, as shown below:

```
tar xf /tmp/rti_file.tar.gz -C /opt/rti
```

- c. Find the shared library directory. The following example identifies the library directory:

```
LIBDIR=$(dirname $(find /opt/rti -name "libRTI-NG.so"))
```

- d. Add the shared library directory to the dynamic linker configuration file:

```
echo $LIBDIR >> /etc/ld.so.conf.d/rti.conf
```

- e. Call **ldconfig** and update the dynamic linker:

```
cache /sbin/ldconfig
```

- f. To save space, delete the original installer that you created in Step 3:

```
rm -f /tmp/rti_file.tar.gz
%end
```

5. Save and close the **hla.ks** script.

B-6 Install TTS software, libraries, and voice data

To save space, diskless images do not include text-to-speech (TTS) software, libraries, and voice data by default. If your application requires TTS, follow these steps to add TTS to a diskless system:

1. On the ACE Studio development workstation, open a terminal, and create a **tts.ks** file for the **ace-tts** RPM and third-party TTS library.
2. In the new **tts.ks** file, install the **ace-tts** RPM:

```
%packages
ace-tts
%end
```

3. Save and close the **tts.ks** file.
4. Unpack the TTS voice data:
 - a. Make a temporary, parent-level directory to store the TTS data:

```
mkdir -p /tmp/tts_data
```

- b. From the ASTi-provided Telestra Text-To-Speech Software Package CD, copy a TTS RPM file into the directory you created in Step 5a:

```
cp TTS_file.rpm /tmp/tts_data
```

where *TTS_file.rpm* represents the TTS file (e.g., **rs-American-English-en-US-Tom-4.5.00-00.i386.rpm**).

- c. Change the working directory to a temporary space:

```
cd /tmp/tts_data
```

- d. Convert the binary .rpm file to a cpio-formatted archive with the **rpm2cpio rs-** command, and unpack the output of this conversion with the **cpio -id** command:

```
rpm2cpio TTS_file.rpm | cpio -id
```

5. Because ACE Studio does not have network share capabilities, it is not suitable to store the TTS voice data long term. As a result, you must copy the TTS directory to a network share on a host that a running, diskless image can mount.

On a Telestra server running ACE Target, run the **scp -r** command to secure-copy the **usr/local/ScanSoft/RealSpeak_4.0** directory to the Target's Network File System (NFS) TTS directory:

```
scp -r usr/local/ScanSoft/RealSpeak_4.0/* xxx.xxx.xxx.xxx:/nfs/tts/
```

where *xxx.xxx.xxx.xxx* represents the network share's IP address (e.g., 192.168.1.1).

6. To log into the diskless image, enter **ssh username@yyy.yyy.yyy.yyy**, where *username* represents your username and *yyy.yyy.yyy.yyy* represents the diskless image's IP address.
7. To test that the network files uploaded correctly, mount the network share:

```
mount xxx.xxx.xxx.xxx:/nfs/tts /usr/local/ScanSoft/RealSpeak_4.0
```

where *xxx.xxx.xxx.xxx* is the network share's IP address.

8. Verify the TTS directory structure exists on the Target:

```
ls usr/local/ScanSoft/RealSpeak_4.0/  
api doc speech
```

Appendix C: 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 start alphabetically. To view all start-up scripts for run level 4, run the `ls -l /etc/rc4.d/S*` command.

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

Script	Function
s10network	This script is the beginning of the diskless network initialization. Before it runs, no network access is available. After it runs, the eth0 network interface is initialized using Dynamic Host Configuration Protocol (DHCP).
s11diskless_network	This script translates the <code>diskless_network.ini</code> global network settings into the specific settings for the individual Telestra Target. After this script executes, all configured network interfaces are available.
s12syslog	Script that runs prior to S12 have no messages logged through syslog .
s13iscsi	For Internet Small Computer Systems Interface (iSCSI) services, configure iSCSI settings before this script runs.
s55sshd	After this service starts, remotely log in or use Secure Shell (SSH) services, such as SSH File Transfer Protocol (SFTP) or Secure Copy (SCP). <i>Note: SSH commands can run on the Target itself after the s11diskless_network script executes.</i>
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 start-up. One script that executes at this priority is the S99 layout. This script installs the default layout for the Target. Any dependencies such as the project, sound files, and default layout selection must be in place before this priority level.

Table 2: Initialization scripts