Dell™ OpenManage™ Deployment Toolkit 3.2

User's Guide

Notes and Cautions

NOTE: A NOTE indicates important information that helps you make better use of your computer.

CAUTION: A CAUTION indicates potential damage to hardware or loss of data if instructions are not followed.

Information in this document is subject to change without notice.
© 2009 Dell Inc. All rights reserved.

Reproduction of these materials in any manner whatsoever without the written permission of Dell Inc. is strictly forbidden.

Trademarks used in this text: *Dell*, the *DELL* logo, *PowerEdge*, *PowerVault*, and *OpenManage* are trademarks of Dell Inc.; *Intel386* is a trademark and *Intel* and *Pentium* are registered trademarks of Intel Corporation in the United States and other countries; *Microsoft*, *MS-DOS*, *Windows*, *Windows NT*, *Active Directory*, and *Windows Server* are either trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries; *Novell*, *NetWare*, and *SUSE* are registered trademarks of Novell, Inc. in the United States and other countries; *Red Hat* and *Red Hat Enterprise Linux* are registered trademarks of Red Hat, Inc. in the United States and other countries.

Other trademarks and trade names may be used in this document to refer to either the entities claiming the marks and names or their products. Dell Inc. disclaims any proprietary interest in trademarks and trade names other than its own.

The sample scripts provided with the Dell OpenManage Deployment Toolkit are provided "as is." Dell disclaims any and all warranties, express, implied or statutory, with respect to the sample scripts, including, but not limited to, the implied warranties of merchantability, fitness for a particular purpose, title and any warranty of non-infringement. Use the sample scripts at your own risk. Dell shall not be liable for any direct or indirect damages incurred in using the sample scripts. In no event shall Dell or its suppliers be responsible for any direct or indirect damages whatsoever (including, without limitation, damages for loss of profits, loss of use, loss of data, business interruption, or other pecuniary loss, nor for punitive, incidental, consequential, or special damages of any kind, under any part of this agreement, even if advised or aware of the possibility of such damage.

December 2009

Contents

1 Introduction 11

Introduction to the DTK User's Guide **12**

What's New **14**

Prerequisites and Requirements **14**

Supported Operating Systems to Run DTK Utilities 15

Supported Systems 15

Supported BIOS Firmware Versions 15

Installation Prerequisites 15

Deployment Prerequisites 16

Deploying Operating Systems **18**

Deployment Methods **18**

DTK Deployment Overview **19**

Preparing for Deployment 19

Deploying 21

Other Dell Documents You Might Need **23**

Other Supporting Documents You Might Need **24**

Obtaining Technical Assistance **24**

2 Choosing a Microsoft Windows PE-Based Deployment Method 25

Choosing a Deployment Method for Microsoft Windows PE **25**

CD-Based Local Deployment 25

CD-Based Local Deployment With Networking Enabled 26

Using a Third-Party Deployment Solution Framework 28

Deployment Matrix (Windows PE) **28**

3 Preparing the Directory Structure for Deployment in Microsoft Windows PE 31

Upgrading Your DTK Version **31**

Obtaining DTK **31**

DTK Contents **34**

System Utilities and Supporting Files 34

Sample Configuration Files 36

Sample Scripts 38

Documentation 39

Drivers **40**

Additional Information 40

Other Supporting Tools and System Files 41

Setting Up a Directory Structure for Deployment **42**

DTK Utilities and Files 43

System-Specific Files 43

Operating System Installation Files 45

Configuring a Source System **50**
RAC Configuration 51

4 Preparing the Script Files for Deployment Using Microsoft Windows PE 53

Using the Sample DTK Scripts to Run a Full Deployment in Windows PE **54**
The Master Batch File (DEPLOY.BAT) 54

Editing the Sample Batch Files to Perform a Complete System Deployment in Windows PE **55**
Using the Sample Scripts in a Scripted Deployment 55
Scripts for Deployment 56
TKENVSET.BAT Example Editing 60
Editing DTK Task Scripts 64

Using the DTK Sample Scripts to Capture Configuration Information in Windows PE **66**
Capturing the System BIOS and BMC Configuration With the SYSCAP.BAT Sample Script 66
Capturing the RAC Configuration With the RACCAP.BAT Sample Script 67

Preparing the Operating System Installation Scripts for Windows PE **69**
WININST.BAT 69
UNATTEND.TXT 70
UNATTEND.XML 72

5 Running Deployment Scripts Using DTK and Microsoft Windows PE 73

Creating and Customizing Images for Windows PE 1.0 . . . . . **73**

Obtaining Windows OEM Preinstallation Kit (OPK) or Windows PE CD 73

Copying Tools From OPK or Windows PE CD on a Development System 73

Creating a Standard Windows PE 74

Integration of DTK Directory Structure in Windows PE Build 75

Creating a Bootable Windows PE ISO Image . . . 76

CD-Based Local Deployment 77

Running the Windows PE 1.0 Image **77**

CD-Based Local Deployment With Networking Enabled 77

Using a Third-Party Deployment Solution Framework 77

Creating and Customizing Images for Windows PE 2.0 . . . . . **78**

Integration of DTK Directory Structure in Windows PE Build 78

Creating a Bootable CD 79

Running the Windows PE 2.0 Image **79**

CD-Based Local Deployment 79

CD-Based Local Deployment With Networking Enabled 80

Using a Third-Party Deployment Solution Framework 80

6 Choosing an Embedded Linux-Based Deployment Method 83

Choosing a Deployment Method for Embedded Linux **83**

Network-Based Deployment 84

Using a Third-Party Deployment Solution Framework 85

CD-Based Local Deployment for Embedded Linux 85

Deployment Matrix (Embedded Linux) **87**

7 Preparing the Directory Structure for Deployment in Embedded Linux 91

Upgrading Your DTK Version **91**

Installation for Embedded Linux **91**

DTK Contents **95**

System Utilities and Supporting Files 95

Sample Configuration Files 97

Sample Scripts 98

Documentation 98

Drivers and Dell Real Mode Kernel (DRMK) 99

Other Supporting Tools and System Files **99**

Setting Up a Directory Structure for Deployment . . . **100**

DTK Utilities and Files 101

System-Specific Files 101

Creating Operating System Installation Directories for Supported Linux Operating Systems 102

Configuring a Source System **104**
How to Obtain Dell Utility Partition Files 105
RAC Configuration 105

8 Preparing the Script Files for Deployment Using Embedded Linux . . . 107

Using the Sample DTK Scripts to Run a Full Deployment in Embedded Linux **107**
The Master Deployment File 107

Editing the Sample Batch Files to Perform a Complete System Deployment in Embedded Linux **108**
Using the Sample Scripts in a Scripted Deployment 108
Editing the tkenvset.sh Script 115
tkenvset.sh 115
Editing DTK Task Scripts 120

Using the DTK Sample Scripts to Capture Configuration Information in Embedded Linux **122**
Capturing the System BIOS and BMC Configuration With the syscap.sh Sample Script 122
Capturing the RAC Configuration With the raccap.sh Sample Script 123

Preparing the Operating System Installation Scripts **125**
Red Hat Enterprise Linux Server Operating Systems 125
lininst.sh 125
ks.cfg and ks-rhel5.cfg 126
Install Linux Using lininst.sh 126

SUSE Linux Enterprise Server Operating Systems 127
suseinst.sh 127
autoinst.xml for SUSE Linux Enterprise Server 129

9 Running Deployment Scripts Using DTK and Embedded Linux 131
Deployment Using Dell-Provided Embedded Linux 131
Deployment Using Customized Embedded Linux 134
Using a Third-Party Deployment Solution Framework 134

10 Running Dell Update Packages in Embedded Linux 135
Running Update Packages in Dell-Provided Embedded Linux 135
Running Update Packages in Customized Embedded Linux 135

11 Known Issues and Frequently Asked Questions 139
Known Issues 139
General Issues 139
RAIDCFG Issues 139
RACADM Issues 140
Windows PE Installation Issues 140
Frequently Asked Questions 140
General Deployment Questions 140

RAIDCFG Questions 141
SYSCFG Questions 145
Embedded Linux Questions 146
Windows PE Questions 147

Index 149

1

Introduction

The Dell™ OpenManage™ Deployment Toolkit (DTK) includes a set of utilities for configuring and deploying Dell systems. The DTK is designed for customers who need to build scripted installations to deploy large numbers of systems in a reliable fashion without having to dramatically change their current deployment processes.

In addition to the command line utilities used to configure various system features, the DTK also provides sample scripts and configuration files to perform common deployment tasks and documentation. These files and scripts describe the use of DTK in Microsoft® Windows® Preinstallation Environment (Windows PE) and embedded Linux environments.

The benefits of the DTK:

- Provides the tools necessary to automate the pre-operating system configuration tasks and the unattended operating system installation tasks when deploying Dell systems.
- Scales to support from one to many system deployment efforts.
- Facilitates consistent system configurations across multiple systems.
- Provides diverse and useful deployment tools that can be utilized in many different ways.

Introduction to the DTK User's Guide

While the DTK is designed to be flexible and expandable, this guide is limited to providing general, best practices procedures that focus on the basic tasks for a successful deployment. Users might find many alternative solutions to the best practices guidelines described in this guide and many additional uses for the DTK utilities beyond what is covered in this guide or in the *Dell OpenManage Deployment Toolkit Command Line Interface Reference Guide*.

CAUTION: The best practices guidelines described in this document were developed and tested on specific systems with specific hardware and software configurations. As a result, the examples and instructions described in this guide are not applicable to every environment. The open nature of the DTK utilities and scripting methodology is designed to allow the flexibility to modify the examples, procedures, and suggestions to best fit your specific environment. Using the DTK examples without first making modifications based on your specific IT environment is likely to result in failed deployments.

Information in this guide is distinctly divided for Windows PE and embedded Linux. Even though the nature of topics covered for both environments is similar, the contents are very different, and has been organized into separate sections. This will enable the user to read only those sections that are relevant. Content in this *User's Guide* is organized as documented in Table 1-1.

Table 1-1. Information Organization in This Guide

Topics	Section for Windows PE	Section for Embedded Linux
Introduction, that includes information on: • DTK • What's new for this release • Prerequisites and requirements • DTK deployment overview	Introduction	Introduction
Planning, that includes best practices for choosing a deployment method	"Choosing a Microsoft Windows PE-Based Deployment Method"	"Choosing an Embedded Linux-Based Deployment Method"

Table 1-1. Information Organization in This Guide *(continued)*

Topics	Section for Windows PE	Section for Embedded Linux
Preparing the deployment directory structure, that includes best practices for: • Installing DTK • Gathering and organizing deployment files (DTK, operating system, drivers, systems management software, and so on) • Preparing an optimally configured source system	"Preparing the Directory Structure for Deployment in Microsoft Windows PE"	"Preparing the Directory Structure for Deployment in Embedded Linux"
Preparing the deployment script files, that includes best practices for: • Editing and modifying sample scripts • Capturing basic input/output system (BIOS), baseboard management controller (BMC), and remote access controller (RAC) configuration profiles from an optimally configured source system	"Preparing the Script Files for Deployment Using Microsoft Windows PE"	"Preparing the Script Files for Deployment Using Embedded Linux"
Running a scripted deployment that includes best practices for: • Creating deployment media • Running the deployment	"Running Deployment Scripts Using DTK and Microsoft Windows PE"	"Running Deployment Scripts Using DTK and Embedded Linux"
Running Dell Update Packages in the embedded Linux environment on supported Dell systems.	Not applicable	"Running Dell Update Packages in Embedded Linux"

NOTE: The best practices examples included in this guide are intended to provide illustrations for using specific scripts and configurations that have been tested on specific Dell systems.

This guide also includes a "Known Issues and Frequently Asked Questions" section.

What's New

The new features for this release include:

- Support for new Dell systems

  NOTE: For the list of Dell systems supported for this version of DTK, see the **readme.txt** file in the root folder, or the *Dell Systems Software Support Matrix* on the *Dell Systems Management Tools and Documentation* DVD or on the Dell Support website at **support.dell.com**.

- Support for new CLI commands. See *Dell OpenManage Deployment Toolkit Command Line Interface Reference Guide* for more information.
- Support for solid state drives.
- Support for 64-bit supported operating system
- Support for Series 7 PERC controllers
- Support for S100 and S300 software RAID controllers

Prerequisites and Requirements

The DTK is a collection of Windows PE- and Linux-based utilities and scripts that assist in performing pre-operating system tasks and deploying Dell systems. An advanced knowledge of Windows PE and Linux is required for users who want to use the DTK utilities to perform pre-operating system configuration tasks or to run a scripted deployment in the respective environments.

A pre-operating system environment is defined as the environment used to configure system hardware *before* a major operating system, such as Windows or Linux, is installed. The DTK utilities and scripts, for example, are executed in a Windows PE or embedded Linux environment for hardware configuration.

Because of the open nature of the utilities and scripts and the environment in which they can be used, it is strongly recommended that DTK users have an advanced understanding of the Windows PE and Linux environment and scripting knowledge for the respective environments. DTK users with an advanced understanding of Windows PE and Linux can take full advantage of the utilities and exploit their capabilities above and beyond what the sample scripts can provide.

CAUTION: Some of the DTK utilities can destroy data if used incorrectly. To avoid the potential risk of data loss, take all necessary precautions to protect data so that mission-critical systems are not disrupted in the unlikely event of a failure. See the *Deployment Toolkit Command Line Interface Reference Guide* for complete information about the capabilities of each DTK utility.

Supported Operating Systems to Run DTK Utilities

For Windows PE, the DTK utilities support Windows PE version 2005 (created using 32-bit versions of Microsoft Windows Server 2003 family with Service Pack 1 [SP1]) and Windows PE 2.0 (created using 32-bit and 64-bit version built on the Windows Vista kernel).

NOTE: DTK utilities do not support Windows PE created with any 64-bit version of Microsoft Windows Server 2003.

For embedded Linux, the DTK utilities support Red Hat® Enterprise Linux® Server (version 4, update 4) 32-bit.

Supported Systems

For the list of Dell systems supported for this version of DTK, see the **readme.txt** in the root folder, or the *Dell Systems Software Support Matrix* on the Dell Support website at **support.dell.com/manuals**.

Supported BIOS Firmware Versions

A minimum BIOS version is recommended on some of the platforms for all the DTK options to work correctly. In general, it is good practice to use the latest ("n") version of BIOS firmware or the "n-1" version that is available on the Dell Support website at **support.dell.com**.

Installation Prerequisites

Before installing the DTK components for Windows PE, ensure that you have the following:

- The DTK self-extracting zip file (**DTK3.X-WINPE-XX.exe**), which contains the utilities, sample scripts, sample configuration files, and documentation you need to deploy your Dell system
- A Windows workstation that has at least 500 MB of free hard-drive space

Before installing the DTK components for embedded Linux, ensure that you have the following:

- The DTK ISO image, which contains the utilities, sample scripts, sample configuration files, and documentation you need to deploy your Dell system
- A Linux workstation that has at least 200 MB of free hard-drive space

Deployment Prerequisites

Before beginning the deployment process for Windows PE, ensure that you have all of the following tools, software, and information ready to use or consult:

- Advanced knowledge of Windows
- OEM Preinstallation Kit (OPK) or Windows PE CD and Windows Server 2003 (with SP1) to build Windows PE 1.0
- Windows Automated Installation Kit (Windows AIK) CD version 936330 or later to customize Windows PE 2.0
- An unzip utility
- Working knowledge to build Windows PE 1.0 and Windows PE 2.0 (See the Microsoft documentation to customize Windows PE 1.0 and Windows PE 2.0)
- Working knowledge of Microsoft Remote Installation Services (RIS) and Automated Deployment Services (ADS) (including setting up of RIS and ADS environments) or any other third party deployment system or tool for Windows PE 1.0
- Working knowledge of Windows Deployment Services (WDS) or any other third party deployment system or tool for Windows PE 2.0
- A workstation with the following capabilities:
  - A writable CD drive and CD-writing software (optional)
  - Network access
- A target system with a CD drive, if performing a local deployment
- A target system with a CD drive and network access, if performing a network deployment
- All DTK utilities, sample scripts, and sample configuration files
- All other utilities and files necessary to perform the deployment, including all required Windows PE drivers, operating system drivers, and the Dell utility partition image file
- DTK documentation:

  - *Dell OpenManage Deployment Toolkit Version Command Line Interface Reference Guide*
  - *Dell OpenManage Deployment Toolkit Version User's Guide*
  - The *Dell Systems Management Tools and Documentation* DVD

NOTE: You can also obtain the latest drivers from the Dell Support website at **support.dell.com.**

- Your operating system software and documentation
- An optimally configured source system with network access

For embedded Linux, ensure that you have all of the following tools, software, and information ready to use or consult:

- Advanced knowledge of Linux and Linux scripting (Bash), Linux networking, installing and working with RPM Package Managers (RPMs), and creating and modifying loop file systems
- Any third party deployment system or tool
- A workstation with the following capabilities:
  - A writable CD drive and CD-writing software (optional)
  - Network access
- A target system with a CD drive, if performing a local deployment
- A target system with a CD drive and network access, if performing a network deployment
- All DTK utilities, sample scripts, and sample configuration files
- All other utilities and files necessary to perform the deployment, including all required Linux drivers, operating system drivers, and the Dell utility partition file
- DTK documentation:
  - *Deployment Toolkit Version Command Line Interface Reference Guide*
  - *Deployment Toolkit Version User's Guide*
- The *Dell Systems Management Tools and Documentation* DVD

NOTE: You can also obtain the latest drivers from the Dell Support website at **support.dell.com.**

- Your operating system software and documentation
- An optimally configured source system with network access

Deploying Operating Systems

After the system hardware is versioned and configured, an operating system can be installed using unattended setup files, samples of which are provided as part of the DTK. While these sample configuration files are operating system specific, they have been enhanced for Dell systems. See your operating system documentation for more information about how to modify these unattended installation files.

For Windows PE, the sample configuration files contain examples to deploy the following operating systems:

- Windows 2008 x64 R2
- Windows Server 2003 with SP1 (32- and 64-bit)
- Windows Server 2003 with SP2 (32- and 64-bit)
- Windows Server 2008 (32- and 64-bit)

NOTE: For the latest supported operating systems, see the Windows PE documentation.

For embedded Linux, the sample configuration files contain examples to deploy the following operating systems:

- Red Hat Enterprise Linux Server version 4 (32- and 64-bit)
- Red Hat Enterprise Linux Server version 5 (32- and 64-bit)
- SUSE® Linux Enterprise Server version 10 (64-bit)
- SUSE® Linux Enterprise Server version 11 (64-bit)

Deployment Methods

Dell systems can be deployed using several different methods. The most common method is using the Systems Build and Update Utility on the *Dell Systems Management Tools and Documentation* DVD that is shipped with your system. Currently, this method is used to deploy a single system at a time with the installation wizard.

The DTK, on the other hand, offers a complete set of utilities and sample scripts that can be used to automate deployment of large numbers of Dell systems. This guide is designed to help guide you through some of the most basic planning considerations, logistical preparations, and deployment procedures to get you started using the DTK to deploy Dell systems.

DTK Deployment Overview

Preparing for Deployment

The deployment process can be separated into two main tasks: preparation and the actual deployment.

To prepare for using the DTK utilities and scripts to deploy to a target system in the Windows PE environment:

1. Copy or extract the DTK utilities, sample scripts, sample configuration files, and drivers provided (in the zip file) into the Windows PE image.
2. Organize the DTK utilities, scripts, and configuration files, the operating system installation files, and the requisite system files and drivers on a network share or local media.
3. Set up an optimally configured source system by using the Dell Systems Build and Update Utility to install your operating system; and the Systems Service and Diagnostics Tools to load drivers. (This source system will be the master server that will be used to replicate settings onto target servers.)

   NOTE: You can also obtain the latest drivers from the Dell Support website at **support.dell.com.**
4. Edit the DTK sample script files that read the system configuration files to set up the system BIOS, BMC, RAID, and RAC and then install an operating system on a target system.
5. Generate a system BIOS, BMC, RAID, and/or RAC configuration profile from the optimally configured source system. (This source system will be the master server that will be used to replicate settings onto target servers.) You can obtain the system BIOS, BMC, RAID, and/or RAC configuration files by booting from the DTK Windows PE ISO image and running the **SYSCAP.BAT**, **RAIDCAP.BAT**, and **RACCAP.BAT** scripts. Then, copy the generated configuration files to a read/write share onto the workstation.

   NOTE: In previous releases, you could use a system profile generated on a specific Dell system to deploy only that same system model. From DTK version 2.4 and later, you can use a system BIOS configuration profile generated for all systems belonging to the same generation. For example, **syscfg_x8xx.ini** for *x8xx* systems, **syscfg_x9xx.ini** for *x9xx* systems, and so on.
6. Create an operating system answer file that contains unattended operating system software installation information.

To prepare for using the DTK utilities and scripts to deploy to a target system in the embedded Linux environment:

1. Obtain the DTK Linux ISO image, which is a self-contained bootable ISO image.
2. Use and customize the sample scripts per your requirements.
3. Set up an optimally configured source system by using the Systems Build and Update Utility on the *Dell Systems Management Tools and Documentation* DVD to install your operating system. Obtain the latest drivers from the Dell Support website at **support.dell.com**.
4. Install an operating system on a target system.
5. Generate a system BIOS, BMC, RAID, and/or RAC configuration profile from the optimally configured source system. (This source system will be the master server that will be used to replicate settings onto target servers.) You can obtain the system BIOS, BMC, RAID, and/or RAC configuration files by booting from the DTK Linux ISO image and running the **syscap.sh**, **raidcap.sh**, and **raccap.sh** scripts. Then, copy the generated configuration files to a read/write share onto the workstation.

   NOTE: In previous releases, you could use a system profile generated on a specific Dell system to deploy only that same system model. From DTK version 2.4 and later, you can use a system BIOS configuration profile generated for all systems belonging to the same generation. For example, **syscfg_x8xx.ini** for *x8xx* systems, **syscfg_x9xx.ini** for *x9xx* systems, and so on.
6. Create an operating system answer file that contains unattended operating system software installation information.

Deploying

After you have prepared all scripts, files, and utilities, you are ready to proceed with the deployment process. While there are many ways to use the DTK utilities to deploy Dell systems, this guide focuses on the two most common methods for Windows PE and embedded Linux: CD-based deployment (local) and network-based deployment (remote).

Deployment Using Removable Boot Media With a Network Connection (CD-Based)

1. Create a deployment CD containing a bootable image with the appropriate CD and network drivers, along with any utilities needed to connect to a network share. The deployment CD initiates the deployment process by mapping to the network share where the deployment directory structure resides.
2. Insert the bootable CD into the appropriate drive of the system to be deployed.
3. Boot or reboot the system.

   The configuration process begins:

   - Windows PE or embedded Linux, as the case may be, is loaded from the CD.
   - Network shares are mapped.
   - The DTK scripts execute the necessary DTK utilities from the network share or the CD itself.
   - Configuration information is read from the network share.
   - The operating system is installed from a network share. (This installation may happen after your system reboots.)

Deployment Using Removable Boot Media Without a Network Connection (CD-Based)

1. Create a deployment CD containing a bootable image with the appropriate drivers. The CD must also include the complete deployment directory structure, which contains all the DTK utilities, scripts, and configuration files, an operating system installation answer file, and the required operating system installation files and drivers.
2. Insert the bootable CD into the CD drive of the system to be deployed (the target system).
3. Boot or reboot the target system.

   The deployment process begins:

   - Windows PE or embedded Linux, as the case maybe, is loaded from the CD.
   - The DTK scripts execute the necessary DTK utilities from the CD.
   - Configuration information is read from the CD.
   - The operating system is installed from the CD.

Network-Based Deployment (Remote)

For Windows PE environments, deployment using RIS is recommended. (For details, see the Microsoft RIS documentation.) You can also use any other deployment tool such as ADS.

For embedded Linux environments, a PXE set up is recommended. (For details, see documentation available on your Linux operating system.)

Other Dell Documents You Might Need

In addition to this *User's Guide* and the *Deployment Toolkit Command Line Interface Reference Guide*, both of which are available as part of the DTK download and on the Dell Support website at **support.dell.com/manuals**, you can find the following guides on the Dell Support website by using the direct link on the *Dell Systems Management Tools and Documentation DVD*:

- The *Dell OpenManage Installation and Security User's Guide* provides additional information about performing an unattended installation of Server Administrator on systems running supported Windows, Red Hat Enterprise Linux Server, and SUSE Linux Enterprise Server operating systems.
- The *Dell OpenManage Legacy Compatibility Guide* provides compatibility information about Server Administrator installation and operation on various hardware platforms (or systems) running supported Windows, Red Hat Enterprise Linux Server, and SUSE Linux Enterprise Server operating systems.
- The *Dell Systems Software Support Matrix* provides information about the various Dell systems, the operating systems supported by these systems.
- The *Dell Systems Build and Update Utility Quick Reference Guide* is an introduction to the Dell Systems Build and Update Utility, and provides tips on how to make the product effective in helping you streamline your deployment and update procedures for Dell systems.
- The *Dell Remote Access Controller 5 User's Guide* provides comprehensive information about using the **RACADM** command line utility to configure a DRAC 5.
- The *Dell Chassis Management Controller User's Guide* provides comprehensive information about using the controller that manages all modules in the chassis containing your Dell system.
- The *Integrated Dell Remote Access Controller User's Guide* provides information about installation, configuration, and maintenance of the Integrated Dell Remote Access Controller (iDRAC) on management and managed systems.
- The *Dell Baseboard Management Controller Utilities User's Guide* provides information about configuring a managed system to use the BMC Management Utility to manage your system through its BMC.

- The *Dell Update Packages User's Guide* provides information about obtaining and using Dell Update Packages as part of your system update strategy.
- For information on terms used in this document, see the *Glossary* on the Dell support website.

Additionally, the Deployment Toolkit **readme.txt** file, which is available as part of the DTK download and on the Dell Support website at **support.dell.com/manuals**, provides the latest available information for the installation and operation of the DTK components.

Other Supporting Documents You Might Need

Besides the Dell-provided documentation, there are numerous other resources to aid you in planning and executing a DTK-assisted deployment:

- Operating system documentation to prepare for and execute the unattended installation process. In addition, you should consult the available Web-based resources such as:
  - The Microsoft Tech Net database at **www.microsoft.com/technet**
  - The Red Hat Enterprise Linux support pages at **www.redhat.com**
  - The SUSE Linux Enterprise Server support pages at **www.novell.com**
- Windows PE 1.0, OEM Preinstallation Kit (OPK), and ADS documentation
- Windows PE 2.0, Windows Automated Installation Kit (Windows AIK), and Windows Deployment Services (WDS) documentation
- Imaging software documentation, if you are creating a bootable CD or are planning to deploy from an image (deploying from an image is not covered in this guide)

Obtaining Technical Assistance

If at any time you do not understand a procedure described in this guide or if your product does not perform as expected, help tools are available to assist you. For more information about these help tools, see "Getting Help" in your system's *Hardware Owner's Manual*. Additionally, Dell Enterprise Training and Certification is available; see **www.dell.com/training** for more information. This service may not be offered in all locations.

2

Choosing a Microsoft Windows PE-Based Deployment Method

A well-thought-out deployment plan is critical to the success of your deployment effort and includes the following considerations:

- Assessing your existing IT environment
- Selecting an operating system
- Selecting the optimum Dell™ OpenManage™ systems management software configuration for your system
- Choosing a deployment method

The Deployment Toolkit (DTK) is designed to be flexible enough to fit into almost any deployment plan. For the purposes of this document, however, the focus is on two specific deployment methods: CD-based deployment (local) and network-based deployment (remote).

Choosing a Deployment Method for Microsoft Windows PE

The DTK components can be used in many different ways to assist IT staff in deploying Dell systems in large numbers. Because all the tools and scripts are task oriented, many of them can be used separately to perform a specific task on many systems or collectively to perform many tasks on many systems at once.

The configuration complexity and pros and cons of each deployment method are provided in Table 2-1. After you have decided which deployment method best fits your needs, you are ready to begin preparing for the deployment.

CD-Based Local Deployment

The general CD-based deployment process includes:

- Creating a bootable Microsoft® Windows® PE CD
- Integrating latest drivers to recognize Dell-specific components
- Integrating DTK and its components (tools and scripts)
- Integrating operating system installation files for deployment

This deployment method is easy to assemble, but is the least flexible. Any change to the system configuration, for example, requires the creation of a new bootable CD. This method does not provide a writable media, making it necessary to use predefined and tested configuration files.

Creating a Customized Bootable Windows PE CD

See "Running Deployment Scripts Using DTK and Microsoft Windows PE" for details on how to create a customized version of a single Windows PE image that can work across all supported Dell systems. You must include Dell-customized drivers and instrumentation drivers into your Windows PE CD-based operating system.

Complete Deployment Using DTK and Operating System Files From a CD Bootable Media

After the customized version of the bootable Windows PE CD is created, IT administrators can take advantage of the sample task-oriented scripts provided in DTK to automate the whole deployment process. For this purely CD-based deployment, it is necessary to add the operating system installation files on the CD media. After the configuration is completed using DTK tools, the operating system can be installed using its native installer in unattended mode.

CD-Based Local Deployment With Networking Enabled

NOTE: The DTK network-based deployment is not supported from mapped Novell® NetWare® systems.

This method provides greater flexibility and is highly recommended in large deployments. The prerequisites are the availability of network bandwidth and that all target systems are connected to the network. A brief description of the tasks for this method is given below.

Bootable Windows PE CD With Networking Enabled

This process includes:

- Creating a bootable Windows PE CD with the appropriate network drivers. This step enables networking services to start, get an IP address, and bring the target system into a functioning network.

  NOTE: See "Running Deployment Scripts Using DTK and Microsoft Windows PE" for details on how to create a customized version of a single Windows PE bootable CD that works across all Dell supported systems and provides an underlying networking stack.

- Creating scripts to automatically *map* to a predefined network share.
- Accessing scripts, configuration files, and operating system installation files from the network.

Preparing and Populating the Network Share

This process includes:

- Creating a network share on a system that is always available over the network for the target systems to be deployed
- Ensuring that a large amount of space is available for storing operating system installation files
- Ensuring that proper permissions are assigned to this share so that the target systems can read files from and write files to the share

This deployment method is also easy to assemble. In addition, it provides greater flexibility in terms of making changes to the scripts and configuration files. Any change to the system configuration, for example, will not require re-creation of the bootable CD. This method also provides access to the remote share as writable media. Hence, the configuration files captured during the deployment automation process can be saved to this remote share.

Using a Third-Party Deployment Solution Framework

You can use the DTK with any existing third-party deployment solution framework that provides a Automated Deployment Services (ADS) booting infrastructure for Windows PE 1.0 or Windows Deployment Services (WDS) booting infrastructure for Windows PE 2.0 that can be used as the transport mechanism for the DTK utilities. Because each third-party deployment framework is unique, however, these solutions fall outside the scope of this document. If you plan to utilize a third-party deployment solution framework, keep in mind that the DTK is a Windows PE-based set of tools and scripts, so the deployment solution framework must also support embedded Windows PE as a pre-operating system environment.

Deployment Matrix (Windows PE)

Depending on the scope, current business needs, network setup, and process, you might select any of the suggested methods of deployment. Table 2-1 outlines three typical methods of deployment and the environment most suitable for each.

Table 2-1. Deployment Method Matrix

Deployment Methods	Complexity	Advantages	Disadvantages
Local deployment (bootable CD)	Relatively simple	• Necessary if a network connection is not available or is too slow. • Can install supported Windows operating systems.	• Any change requires the creation of a new bootable CD. • Limited space on CD for some operating system installation files. • Deployment tasks must be performed at the individual target system. • For Windows PE 1.0, CDs are read-only media and require the creation of a RAMDISK for temporary data storage. • Installation time is high.

Table 2-1. Deployment Method Matrix *(continued)*

Deployment Methods	Complexity	Advantages	Disadvantages
Remote deployment over a network (bootable Windows PE CD with network stack loaded)	Moderate	• Everything is in one place in a network share. • Easy to manage—changes have to be made in only one place. • Data captured (profiles) can be stored in a network share. • Data can be replicated from a network share. • Can install supported Windows operating systems.	• Must have network connection. • Deployment tasks must be performed at the individual target system. • Must locate the appropriate Windows PE network drivers. • Not supported from a mapped NetWare system.
Deployment solution framework from a third-party vendor	Mixed	• The DTK can be used in the context of third-party deployment solution framework. • Third-party deployment solution framework is used as the deployment transport mechanism. • Tasks and scripts can be pushed to the target systems.	Must either acquire and learn to use or have an existing third-party deployment solution framework.

After you have determined which deployment method best suits the needs of your organization, you are ready to begin building the DTK deployment directory structure on a network volume. For instructions on installing the DTK components and populating the deployment directory structure, see "Preparing the Directory Structure for Deployment in Microsoft Windows PE."

3

Preparing the Directory Structure for Deployment in Microsoft Windows PE

This section describes some of the procedures necessary to begin the deployment process, including:

- Downloading and unzipping the Deployment Toolkit (DTK) components, including a comprehensive matrix of the DTK components
- Setting up a directory structure to facilitate a successful deployment, including a list of the supporting components that are not included with the DTK, but are needed to use the DTK
- Setting up an optimally configured source system
- Configuring supported RAID controllers

Upgrading Your DTK Version

When upgrading from a previous release (2.*x*) of the DTK, ensure that you create the new DTK version 3.*x* environment separately from your previous version of DTK. This step is necessary because the environments and requirements for DTK version 3.*x* are completely different.

Obtaining DTK

The DTK components are provided in a self-extracting zip file on the Dell™ Support website at **support.dell.com.** The self-extracting file can be opened on any system running a Microsoft® Windows® operating system. By default, the **DTK3.X-WINPE-XX.exe** files are extracted to the root directory of your local hard drive, **C:**. This location can be changed by giving a different path when extracting the files, but it is strongly recommended that you keep the DTK default directory structure if you plan to use the sample deployment scripts provided with the DTK.

Perform the following steps to extract the DTK components to a workstation running Windows:

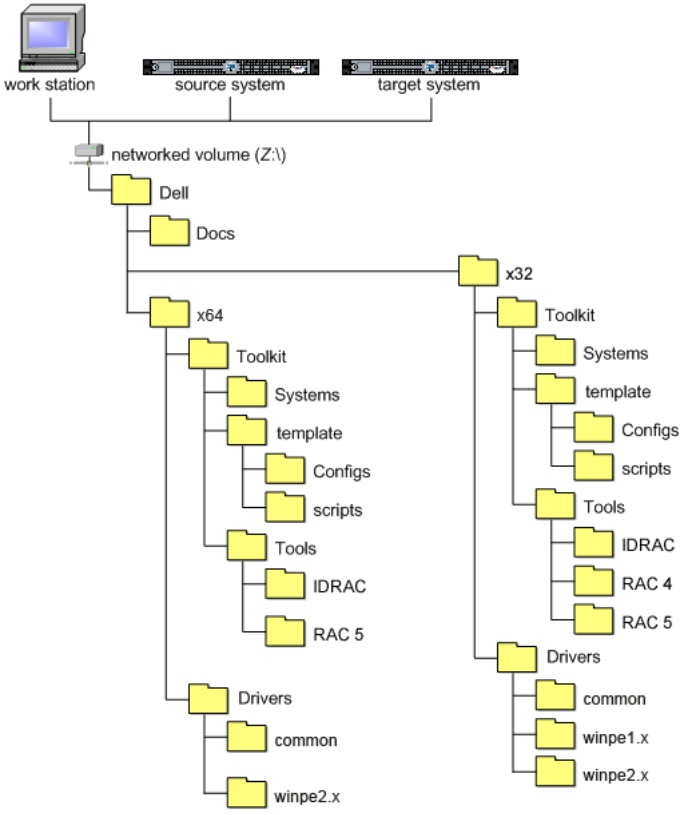
1. Download the DTK file **DTK3.X-WINPE-XX.exe** from the Dell Support website at **support.dell.com** and save it anywhere on your system.
2. After the file downloads, double-click the file.
3. Click **OK**.
4. Click **Unzip**.

   By default, files are unzipped to **C:**. You can unzip the DTK components to your local hard drive and then copy them later to a network volume, or you can change the installation path to an available network share. For example, you can unzip the files directly to the network share (for example, **Z:**) that you plan to use in the deployment process.

   NOTE: Do not copy the DTK components to a Novell® NetWare® network volume. The DTK network-based deployment is not supported from mapped Novell NetWare systems. The DTK utilities do not run in Windows PE when Novell's Client32 is loaded.

The files are unzipped in the directory structure shown in Figure 3-1.

Figure 3-1. Sample DTK Default File Structure



Workstation is the system on which the DTK components are extracted.

Source system is an optimally configured system from which the system settings are captured.

Target system is the system on which the settings captured from the source system are replicated.

DTK utilities for 32-bit systems are located in the following directories:

- Utilities: **Z:\Dell\x32\Toolkit\Tools**
- Sample scripts: **Z:\Dell\x32\Toolkit\template\scripts**
- Sample configuration files: **Z:\Dell\x32\Toolkit\template\Configs**

DTK utilities for 64-bit systems are located in the following directories:

- Utilities: **Z:\Dell\x64\Toolkit\Tools**
- Sample scripts: **Z:\Dell\x64\Toolkit\template\scripts**
- Sample configuration files: **Z:\Dell\x32\Toolkit\template\Configs**

The directory **Z:\Dell\Toolkit\Systems** is a placeholder that can be used for all system information that is captured using sample scripts and tools (for example, the system specific basic input/output system (BIOS) and baseboard management controller (BMC) configuration files used by the DTK utilities for configuring a system).

See "Setting Up a Directory Structure for Deployment" for further discussion of the deployment directory structure.

DTK Contents

The DTK contains Windows PE-based utilities, sample scripts, sample configuration files, drivers, and documentation for automating the deployment of Dell systems. The following sections provide a complete list of the DTK contents and a short description of each component. For a more complete explanation of the DTK utilities, see the *Command Line Interface Reference Guide*. For an explanation of the sample configuration files and scripts, see "Editing the Sample Batch Files to Perform a Complete System Deployment in Windows PE."

System Utilities and Supporting Files

Table 3-1 describes the files, located in **Toolkit\Tools**. These are the core tools (utilities) that perform pre-operating system configuration and deployment tasks. See the *Command Line Interface Reference Guide* for complete information about the capabilities of each DTK utility.

Table 3-1. System Utilities and Supporting Files

Toolkit Components	Purpose
config.ini	Configuration file used by **RAIDCFG.EXE**. **NOTE:** Do not edit this file. Incorrect edits might cause **RAIDCFG.EXE** to fail.
mr2kserv.exe	Service that needs to be running for **RAIDCFG.EXE** to function.
IDRAC	Configures iDRAC6. **NOTE:** This feature may not be supported on all systems.
pci.ids	Used with **SYSCFG.EXE** to report PCI devices.
RAC 4	Configures Dell Remote Access Controller (DRAC) 4/P and DRAC 4/I.
RAC 5	Configures DRAC 5/iDRAC.
racadmErrorCodes.txt	Lists the error codes and messages for the **RACADM.EXE** tool.
RAIDCFG.EXE	Creates and reports Redundant Array of Independent Disks (RAID) information.
RaidcfgErrorCodes.txt	Lists the error codes and messages for the **RAIDCFG.EXE** tool.
stsvc.ini	Configuration file used by **RAIDCFG.EXE**. **NOTE:** Do not edit this file. Incorrect edits might cause **RAIDCFG.EXE** to fail.
sys.ini	A **.ini** file used with **syscfg**.
SYSCFG.EXE	• Reports and configures: – BIOS settings and options – BMC settings and options – RAC 5 settings and options • Maintains system state between reboots • Reports hardware devices and other system-related information
SyscfgErrorCodes.txt	Lists the error codes and messages for the **SYSCFG.EXE** tool.

Table 3-1. System Utilities and Supporting Files *(continued)*

Toolkit Components	Purpose
sysdrmk.exe	Used by **UPINIT.BAT** to patch the boot sector and Master Boot Record (MBR).
UpinitErrorCodes.txt	Lists the error codes and messages for **UPINIT.BAT**.
version.txt	Contains the DTK version in use.

Sample Configuration Files

Table 3-2 describes examples of configuration files used to perform a scripted deployment. The files for 32-bit and 64-bit supported systems are located in the following directories:

- **Z:\Dell\x32\Toolkit\template\Configs**
- **Z:\Dell\x64\Toolkit\template\Configs**

NOTE: All configuration files are provided as examples only. These sample configuration files must be edited before they can be used in any deployment tasks.

Table 3-2. Sample Configuration Files

Toolkit Components	Purpose	Location
RACCFG.INI	Sample configuration file for DRAC 4 remote access controllers.	Move to the directory **\Toolkit\Systems\<system>**.
RAIDCFG.INI	Sample output file to enable RAID replication.	Move to the directory **\Toolkit\Systems\<system>**.
SYSCFG.INI	Sample configuration file for SYSCFG utility to configure the BIOS and BMC settings.	Move to the directory **\Toolkit\Systems\<system>**.

Table 3-2. Sample Configuration Files *(continued)*

Toolkit Components	Purpose	Location
UNATTEND.TXT	Sample file used in Windows PE 1.0 environment with unattended installation of Microsoft Windows operating systems. This file is populated with all the necessary information for Dell-supported mass storage devices.	Move to the directory **\Toolkit\Systems\<system>** after the **UNATTEND.TXT** file is edited.
UNATTEND.XML	Sample file used in Windows PE 2.0 environment with unattended installation of Microsoft Windows operating systems. This file is populated with all the necessary information for Dell-supported mass storage devices.	Move to the directory **\Toolkit\Systems\<system>** after the **UNATTEND.TXT** file is edited.
STARTNET.CMD	Sample script used by Windows PE 2.0 when it boots up.	**\Toolkit\template\Configs**
WINBOM.INI	Sample file used to start the racsvc and mr2kserv services.	**\Toolkit\template\Configs**
WINPEOEM.SIF	Sample file to add support for Dell mass storage drivers.	**\Toolkit\template\Configs**

Sample Scripts

Table 3-3 lists sample scripts of those necessary to perform a scripted deployment. The sample scripts are located in **Toolkit\template\scripts.**

NOTE: All DTK scripts are provided as examples only. The batch scripts and configuration files must be modified to reflect the unique information for each deployed system.

Table 3-3. Sample Scripts

Toolkit Components	Purpose
createup.cfg	This file is used by **UPINIT.BAT**.
DISKPARTCLR.CFG	Used by **PARTCFG.BAT** to clear the partitions on the selected disks.
DISKPARTOS.CFG	Used by **PARTCFG.BAT** to create deployment partition to install the operating system.
ERRHNDL.BAT	Handles errors returned in the DTK scripts.
LISTUP.CFG	This file is used by **UPINIT.BAT**.
MOUNTUP.CFG	This file is used by **UPINIT.BAT**.
PARTCFG.BAT	Creates and populates the Dell utility partition and creates the operating system partition on a specified hard drive. Before doing so, however, this script cleans out all existing partitions in your system.
RACCAP.BAT	Captures remote access controller (RAC) settings to a file for DRAC 4 remote access controllers.
RACREP.BAT	Applies (replicates) the RAC settings saved in a file to the target system.
RAIDCFG.BAT	Configures the selected RAID controllers on the system.
SYSCAP.BAT	Captures BIOS and BMC configuration settings and saves them to a file.
SYSREP.BAT	Applies (or replicates) BIOS and BMC configuration settings and configures the target system.
TKENVSET.BAT	Sets the path of the DTK installation and variables for the specific system being deployed.

Table 3-3. Sample Scripts *(continued)*

Toolkit Components	Purpose
UMOUNTUP.CFG	This file is used by **UPINIT.BAT**.
UPINIT.BAT	Creates and populates the Dell utility partition.
WININST.BAT	Performs an unattended installation of a supported Microsoft Windows 2000 and Windows Server® 2003 operating systems.
W2K8INST.BAT	Performs an unattended installation of supported Microsoft Windows Server 2008 operating system.
DTKRUNALL.BAT	Applies BIOS, BMC, RAC, and RAID settings saved in the files to the current system.

Documentation

Table 3-4 describes documents containing reference information for each DTK utility and instructions for using the DTK utilities and sample scripts. Documentation is located in the following directory: **Z:\Dell\Docs**.

Table 3-4. Documentation

Toolkit Components	Purpose
Command Line Interface Reference Guide	Describes all DTK utilities, including all command line arguments and values, syntax, help, and examples.
User's Guide	Contains best practices guidelines for performing a scripted deployment.

Drivers

Table 3-5 describes the essential and optional drivers required for Dell-supported hardware to build your Windows PE environment.

NOTE: The drivers provided have been tested successfully in the Windows PE environment built during the test.

Table 3-5. Drivers

Drivers	Purpose
winpe1.*x*\ DRIVERINST.BAT	Used to install Dell-provided drivers, from **Z:\Dell\x32\Drivers** or **Z:\Dell\x64\Drivers**, into the customized Windows PE 1.0 image.
winpe2.*x*\ VPE_DRIVERINST.BAT	Used to install Dell-provided drivers, from **Z:\Dell\x32\Drivers** or **Z:\Dell\x64\Drivers**, into the customized Windows PE 2.0 image.
common\Hardware application programming interface (HAPI)	Essential drivers for the DTK tools to work.
winpe1.*x*\Mass Storage	Essential drivers for mass storage controllers to work.
winpe2.*x*\Mass Storage	Essential drivers for mass storage controllers to work.
winpe1.*x*\Network interface card (NIC)	Recommended drivers for NICs.
winpe2.*x*\NIC	Recommended drivers for NICs.
common\RAC 4	Essential drivers for RACADM to work.
common\RAIDCFG	Essential drivers for RAIDCFG to work.

Additional Information

The document **readme.txt**, located at **Z:\Dell\x64** or **Z:\Dell\x32** provides the latest product information.

Other Supporting Tools and System Files

While the DTK provides all the necessary utilities and scripts for deploying Dell systems, there are additional applications, utilities, and system files needed to create the pre-operating system environment in which this toolkit is used. Table 3-6 lists the system files and other tools for you to be able to use the DTK utilities and its sample scripts.

NOTE: Ensure that the tools are 32-bit compatible.

Table 3-6. Supporting Tools and System Files

Additional Tools	Purpose	Source	Save in DTK location
OEM Preinstallation Kit (OPK) or Windows PE CD and Microsoft Windows Server 2003 (with SP1) to build Windows PE 1.0	Create a bootable CD for Windows PE 1.0. **NOTE:** Contact Microsoft to obtain the latest version of OPK or Windows PE CD.	www.microsoft.com	Save in the root directory of the bootable CD.
Windows Automated Installation Kit or Windows PE CD and Microsoft Windows Server® 2003 (with SP1) (Windows AIK) CD to build Windows PE 2.0	Create a bootable CD for Windows PE 2.0. **NOTE:** Contact Microsoft to obtain the latest version of AIK or Windows PE CD.	www.microsoft.com	Save in the root directory of the bootable CD.

Table 3-6. Supporting Tools and System Files *(continued)*

Additional Tools	Purpose	Source	Save in DTK location
upimg.bin	Contains system-specific Dell utility partition files. **NOTE:** In the previous releases, each Dell system required a unique **upimg.bin** file. Starting in DTK 2.3, the **upimg.bin** file is common for all Dell systems.	**\Toolkit\Systems,** the Dell Support website at **support.dell.com,** or the Dell Systems Build and Update Utility on the *Dell Systems Management Tools and Documentation* DVD	Save in the **\Toolkit\Systems** directory**.**

Setting Up a Directory Structure for Deployment

To perform a DTK-enabled deployment, you must create a deployment directory structure on either a network share or your local workstation hard drive. The deployment directory structure is a central repository for all deployment files and can be used successively as a testing space, a repository for a network deployment, and a repository to build a bootable deployment CD. If you plan to perform a network-based deployment, you must set up the directory structure on a network volume that is accessible from your workstation, the source system (to generate configuration information), and the target system (to deploy).

If you plan to run a scripted deployment from the DTK sample scripts, ensure that the following files are correctly structured in the deployment directory. The DTK sample scripts use this structure to set the default paths for deployment. If you make changes to the directory structure, you must also make changes to the sample scripts. The deployment directory structure consists of the following files:

- DTK utilities and files
- System-specific directories
- Operating system installation files directories

NOTE: Do not create your DTK directory structure on a NetWare network volume. The DTK network-based deployment is not supported from mapped NetWare systems.

DTK Utilities and Files

All necessary DTK files are provided as part of the initial installation. It is strongly suggested that you use the default directory structure created in the initial installation of the DTK components. The sample scripts provided with the DTK are set up to use these default paths. See Figure 3-1.

NOTE: If you choose to create a directory structure for the DTK components that is different from the one provided in the initial installation, carefully review and edit each sample script (where necessary) to reflect the different directory structure.

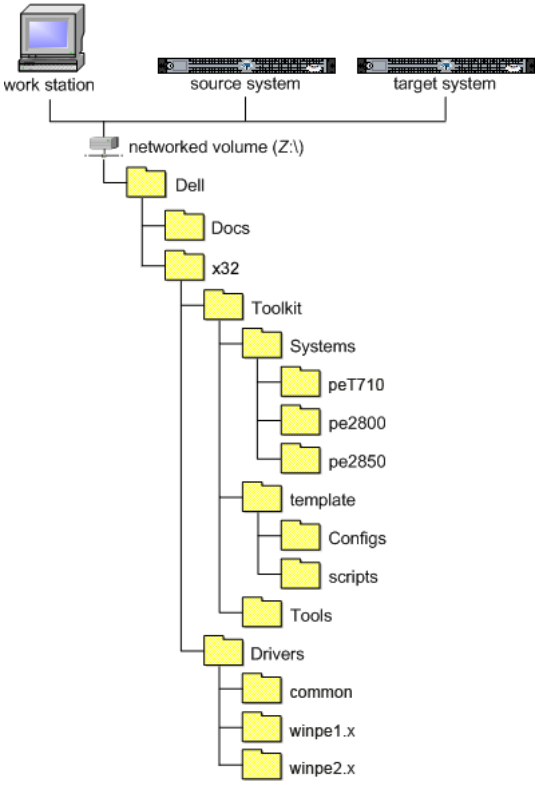
A 32-bit unzip utility is needed to create the Dell utility partition. Ensure that you copy this utility into one of the path directories of your Windows PE 1.0 image, preferably **\I386\system32**; for Windows PE 2.0, the path directory should preferably be **\windows\system32**.

System-Specific Files

The directory **Toolkit\Systems** acts as a central repository for all the system-specific deployment information. For example, you can store all profile configuration files for an optimally configured PowerEdge™ 2850 in the directory **Toolkit\Systems\pe2850** as shown in the figure. These files can then be accessed from this directory when deploying multiple PowerEdge 2850 systems.

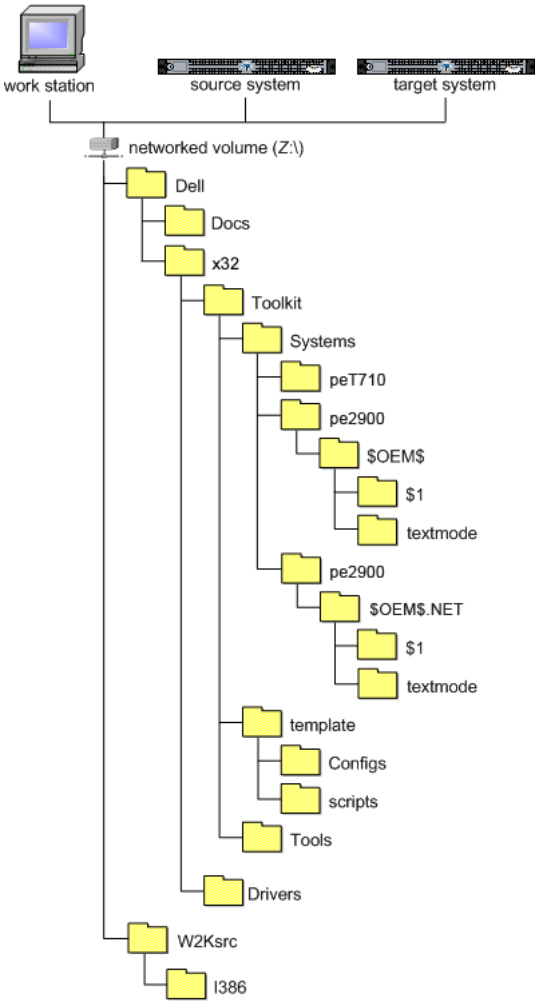
NOTE: If you choose to create a directory structure for the system-specific files that is different from the one suggested in this guide, be sure to carefully review and edit each sample script (where necessary) to reflect the different directory structure.

Figure 3-2. DTK System-Specific File Structure



Operating System Installation Files

Figure 3-3. Sample Operating System-Specific File Structure (Windows Server 2003)



Creating Operating System Installation Directories for Windows Server 2003 Operating Systems

For deployment of Windows Server 2003 operating systems, perform the following steps:

1. On a management station create a folder, for example **os_deploy**.
2. Make this folder shareable.
3. Create two folders **Dell** and **W2K3SRC** under this shareable directory **os_deploy**.
4. Copy the **Dell** folder from WinPE DTK CD to **Dell** folder created under **os_deploy** folder on the management station.
5. Copy **i386** folder from the Windows Server 2003 operating system CD to the **W2K3SRC** folder created under the **os_deploy** folder on the management station.
6. Inside the **os_deploy\Dell\toolkit\systems** folder, create a target system folder for example "**pe4600**". Inside this folder create a folder "**OEM.NET**" for the Windows Server 2003 operating system installation.

   The target system name should be the DTK standard name. To know the system name, run **TKENVSET.BAT** located in the **x:\Dell\Toolkit\template\scripts** directory, and then run the **SET** command. In the **Details** section, see the **dt_platform** name.
7. Create one more directory **textmode** under **OEM.NET**. Copy the extracted controller drivers downloaded from **support.dell.com** to the **textmode** folder.
8. Copy **unattend.txt** from **os_deploy\dell\toolkit\template\configs** folder to **os_deploy\dell\toolkit\systems\pe4600**.
9. Edit **unattend.txt** and modify the following mandatory fields:

Table 3-7. Mandatory Fields in unattend.txt

Field	Description	Example
OEMBootFiles	List of files in the driver folder.	For example, in a system that has PERC 5, you must retain the following entry and remove the rest of the entries under the field OEMBootFiles. DELL_PERC5_PERC6_W2K3 percsas.sys oemsetup.inf nodev.inf percsas.cat
MassStorageDrivers	Storage controller details	For example, in a system that has PERC 5, you must retain the following entry and remove the rest of the entries under the field MassStorageDrivers. DELL PERC5 RAID Controller Driver (Server 2003 for x64)"="OEM"

10 Boot the server with the WinPE DTK media.

11 After WinPE comes up make sure that the server is able to ping the management station.

12 Run the following command to connect the server to the shared folder **os_deploy**.

```
Net use z:\\<share>\os_deploy
```

13 On the server, set the following environment variables:

```
DT_OSSRC=z:\\<share>\W2K3SRC\
```

```
DT_UNATTEND=
z:\\<share>\dell\toolkit\systems\pe4600
\unattend.txt
```

```
DT_OEM=
z:\\<share>\dell\toolkit\systems\pe4600\$OEM$.NET
```

14 Navigate to the mount point **z:\\<share>\dell\toolkit\template\scripts**, and run the **partcfg.bat** script to create the partition and then run the following to install the operating system:

```
wininst.bat NET
```

15 Reboot the server once you are at the command prompt and the system starts booting to the hard drive for operating system installation.

Creating Operating System Installation Directories for Windows Server 2008 Operating Systems

NOTE: The following procedures are provided as a best practices guideline for creating operating system distribution folders compatible with the DTK sample scripts. For complete instructions on setting up an installation directory structure for an unattended installation, see your operating system documentation.

For deployment of Windows Server 2008 operating systems, perform the following steps:

1 Create an operating system-specific directory in the root directory of the network share. For example:
W2K8_deploy

2 Create two directories under the **W2K8_deploy** directory: **sources** and **<platform name>**, for example, **PE2950**.

3 Copy the contents of the operating system CD to **sources**.

4 Under **PE2950**, create two directories: **OEM** and **Out-of-Box Drivers**.

5 Copy all the drivers to **W2K8_deploy\PE2950\ Out-of-Box Drivers**. You can copy each driver to a different sub-folder.

6 Copy **unattend.xml** from **\Toolkit\template\Configs** to **W2K8_deploy\PE2950**.

7 If required, modify the **unattend.xml** for **DiskID** under the tags **DiskConfiguration** and **ImageInstall**. Default value is 0 for **DiskID**.

8 Make sure that **DiskID** specified in the **unattend.xml** is actually available to install the operating system. You can run the **DiskPart** utility to check disk details.

9 Boot your system with Windows PE 2.0 DTK media.

10 Use the following command syntax to connect your system to the shared folder, **W2K8_deploy**:

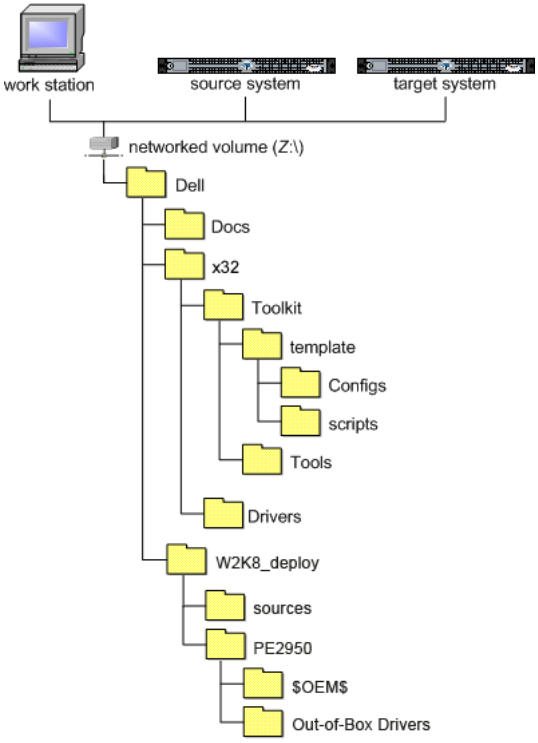
```
Net use  Z:   \\10.94.140.12\W2K8_deploy
```

11 From your system, run **TKENVSET.BAT** under **\Toolkit\template\scripts**

12 On your system, set `DT_DRIVE=Z:`

13 Navigate to **Toolkit\template\scripts** and run **W2K8INST.BAT**.

Figure 3-4. Sample Operating System-Specific File Structure (Windows Server 2008)



Configuring a Source System

You need an optimally configured system both for testing purposes and as the source for the SYSCFG and RAC configuration files to be used in the deployment process. See "Using the DTK Sample Scripts to Capture Configuration Information in Windows PE" for instructions on using the sample DTK scripts, **SYSCAP.BAT** and **RACCAP.BAT**, to generate SYSCFG and RAC configuration files that are used to configure the target system.

You can use the Systems Build and Update Utility and Systems Service and Diagnostics Tools available on the *Dell Systems Management Tools and Documentation* DVD supplied with your system to streamline the operating system installation; to setup and configure your system, including automatic discovery and configuration of Dell-provided RAID controllers and network adapters; to access and load the latest utilities, drivers, and diagnostics for your system; and to install Dell OpenManage™ systems management software.

NOTE: You can also obtain the latest drivers from the Dell Support website at **support.dell.com.**

For complete information about using the Systems Build and Update Utility and Systems Service and Diagnostics Tools, see the *Dell OpenManage Software Quick Installation Guide*, which is available on the Dell Support website at **support.dell.com/manuals**.

After you have loaded the operating system and all applicable device drivers from the Systems Build and Update Utility and Systems Service and Diagnostics Tools, check the Dell Support website at **support.dell.com** for the latest drivers. Put all necessary drivers to be used in the deployment in the system directory for the target system. For example, if your target system is a PowerEdge 1850, put all necessary files in the **\Toolkit\Systems\pe1850** directory.

RAC Configuration

Configure the RAC on your source system using the **RACADM.EXE** utility, if applicable. See the *Dell Remote Access Controller Racadm User's Guide* for comprehensive information about using the **RACADM.EXE** utility to configure the RAC on your source system.

After you have successfully prepared your directory structure for a deployment and fully configured your source system, you are ready to write, review, or edit the deployment sample scripts. For instructions on editing the sample scripts necessary to run a full deployment, see "Preparing the Script Files for Deployment Using Microsoft Windows PE."

4

Preparing the Script Files for Deployment Using Microsoft Windows PE

This section describes the steps required to use the Deployment Toolkit (DTK) sample scripts to perform a full deployment of a target system in the Microsoft® Windows® PE environment.

There are four main tasks:

- Writing the deployment master batch file (as an example, referred to as **DEPLOY.BAT** in this document) or deployment script, editing the **TKENVSET.BAT** script, to set the variable values used in the deployment scripts
- Editing the supporting task scripts, as necessary
- Using the DTK sample scripts to capture basic input/output system (BIOS), baseboard management controller (BMC) (SYSCFG), and remote access controller (RAC) configuration information from an optimally configured source system
- Preparing the operating system installation scripts and answer files

Using the Sample DTK Scripts to Run a Full Deployment in Windows PE

The Master Batch File (DEPLOY.BAT)

A scripted DTK deployment relies on the master batch file (**DEPLOY.BAT**) to integrate all the applicable DTK task scripts and utilities and perform pre-operating system configuration tasks before launching the operating system installation files. **DEPLOY.BAT**, along with the **TKENVSET.BAT** scripts, makes up the master set of instructions that drives the deployment process. This file is not provided with the sample scripts. A sample **DEPLOY.BAT** is given below:

```
::
*****************************************************
::Call TKENVSET.BAT to discover platform information
::as well as set paths and environment variables for
::Deployment Toolkit Tools and Scripts.
*****************************************************
*****************************************************
:: call RAIDCFG.BAT to configure RAID.
::
*****************************************************
*****************************************************
:: Replicate the system settings to the current SUT!.
CALL SYSREP.BAT or DTKRUNALL.BAT.
::
*****************************************************
*****************************************************
::Partitioning the system hard disk. call PARTCFG.BAT.
*****************************************************
```

```
*****************************************************
:: Put hdd as first in boot sequence (1-removable, 2-
cdrom, 3-hdd)
:: If the system configuration has changed (i.e. a
::boot device was added, removed, or enabled in BIOS)
::the following line should be modified to reflect the
::new hdd device number.
:: This is calling Windows installer scripts.
*****************************************************
::%DT_TOOLS%\syscfg --bootseq=hdd.emb.0
::call WININST.BAT NET
```

Editing the Sample Batch Files to Perform a Complete System Deployment in Windows PE

Using the Sample Scripts in a Scripted Deployment

The sample scripts utilize the DTK utilities, the additional supporting utilities, the source system configuration files, the target system hardware drivers, and the operating system installation files to perform a full deployment of the target system.

The sample scripts provided with the DTK can be edited and customized using any ASCII text editor that conforms to the hardware available on the target systems. While it is recommended that you use the DTK sample scripts to automate your deployment tasks, you can create your own system deployment batch files using any ASCII text editor by incorporating the various DTK utilities and commands in the batch file.

NOTE: For more information about scripting batch files and specific batch file commands, see your Windows PE documentation.

Each of the utilities, scripts, and system configuration files are described below. Depending on your requirements, you may use all or only some of them in your deployment:

DEPLOY.BAT— You can write your master batch file (**DEPLOY.BAT**) to perform a full deployment of a system with the help of utilities, scripts, and configuration files. This file is not provided with the sample scripts.

Scripts for Deployment

You may need the following scripts for deployment:

Table 4-1. Capture Scripts

Script Name	Description	Associated Files
RACCAP.BAT	Captures RAC settings to a file for DRAC 4.	• **TKENVSET.BAT** — Used by **RACCAP.BAT** to inform the task scripts where to find the DTK utilities, scripts, and configuration files. • **RACADM.EXE** — Used by **RACCAP.BAT** to retrieve RAC information and to configure settings.
SYSCAP.BAT	Captures BIOS and BMC configuration settings and saves them to a file.	• **TKENVSET.BAT** — Used by **SYSCAP.BAT** to inform the task scripts where to find the DTK utilities, scripts, and configuration files. • **SYSCFG.EXE** — Used by **SYSCAP.BAT** to configure the BIOS and BMC settings.
RAIDCAP.BAT	Captures the RAID settings and saves them to a file.	• **TKENVSET.BAT** — Used by **RAIDCAP.BAT** to inform the task scripts where to find the DTK utilities, scripts, and configuration files. • **RAIDCFG.EXE** — Used by **RAIDCAP.BAT** to retrieve RAID information and to configure settings.

Table 4-2. Replication Scripts

Script Name	Description	Associated Files
RACREP.BAT	Applies the RAC configuration settings saved in a file to the current system.	• **TKENVSET.BAT** — Used by **RACREP.BAT** to inform the task scripts where to find the DTK utilities, scripts, and configuration files. • **RACADM.EXE** — Used by **RACREP.BAT** to replicate RAC information and to configure settings.
RAIDREP.BAT	Applies the RAID settings saved in a file to the current system.	• **RAIDCFG.EXE** — Used by **RAIDREP.BAT** to retrieve RAID information and to configure settings.
SYSREP.BAT	Applies the BIOS and BMC settings saved in a file to the current system.	• **TKENVSET.BAT** — Used by **SYSREP.BAT** to inform the task scripts where to find the DTK utilities, scripts, and configuration files. • **SYSCFG.EXE** — Used by **SYSREP.BAT** to configure the BIOS and BMC settings.

Table 4-3. Deployment Scripts

Script Name	Description	Associated Files
WININST.BAT	Performs an unattended installation of supported Windows 2000 Server and Windows Server® 2003 operating systems.	• **TKENVSET.BAT** — Used by **W2K8INST.BAT** to inform the task scripts where to find the DTK utilities, scripts, and configuration files. • **SETUP.EXE** — Used by **W2K8INST.BAT** to install the operating system in conjunction with the **UNATTEND.XML** answer file. • **UNATTEND.XML** — Used by **W2K8INST.BAT** in conjunction with **SETUP.EXE** to install the operating system files.

Table 4-4. Others

Script Name	Description	Associated Files
TKENVSET.BAT	Sets the path of the DTK installation and variables for the system to be deployed and configured. The variables and paths specified in this file must be specified before any of the other supporting scripts can be used.	• **SYSCFG.EXE** — Used by **TKENVSET.BAT** to discover the system type.

Table 4-4. Others *(continued)*

Script Name	Description	Associated Files
RAIDCFG.BAT	Configures the selected Redundant Array of Independent Disks (RAID) controller detected in the system.	• **TKENVSET.BAT** — Used by **RAIDCFG.BAT** to inform the task scripts where to find the DTK utilities, scripts, and configuration files. • **RAIDCFG.EXE** — Used by **RAIDCFG.BAT** to retrieve RAID information and to configure settings.
PARTCFG.BAT	Creates and populates the Dell™ utility partition (UP) and the operating system partition on a specified disk.	• **TKENVSET.BAT** — Used by **PARTCFG.BAT** to inform the task scripts where to find the DTK utilities, scripts, and configuration files. • **DISKPART.EXE** — Provided by Windows; this script is used to create partitions on your disk. When creating a Windows PE image, ensure that **DISKPART.EXE tool** is present in the Windows PE image you created. This script uses the following files for noninteractive (unattended) operations: • **UPINIT.BAT**— Used by **PARTCFG.BAT to** create and populate the Dell utility partition. • **FORMAT.EXE** — Provided by Windows to format the partition.

TKENVSET.BAT Example Editing

The majority of the editing required for the DTK sample scripts involves the following script. As a result, the majority of the scripting discussion focuses on:

TKENVSET.BAT script**.** Other task scripts might require edits in certain situations. These optional edits are discussed in "Editing DTK Task Scripts."

The **TKENVSET.BAT** script sets the path of the DTK installation and variables for the system to be deployed and configured. The variables and paths specified in this file must be specified before any of the other supporting scripts can be used.

Specifying DTK Installation Paths

The **TKENVSET.BAT** sample script is provided with the following default values. These values must be confirmed, in some cases, and modified, in other cases, before a deployment. Locate the **Deployment Toolkit Installation Paths** heading to find the following lines:

```
set DT_DRIVE=X:

set DT_PATH=%DT_DRIVE%\Dell\Toolkit

set DT_TOOLS=%DT_PATH%\Tools

set DT_SYSTEMS=%DT_PATH%\Systems

set DT_SCRIPTS=%DT_PATH%\Template\Scripts

set DT_STOP_ON_ERROR=FALSE
```

Perform the following steps to confirm and/or edit the installation path variables:

1. Set the variable that represents the drive letter where the DTK deployment files are located (`DT_DRIVE`). This variable is set to `X:` by default in the sample script. Edit this drive letter to reflect the actual drive letter for the DTK deployment files, if necessary.
2. Set the variable that defines the path to the full set of DTK deployment files (`DT_PATH`). This variable is set to `%DT_DRIVE%\Dell\Toolkit` by default in the sample script. Edit this path to reflect the actual location of the DTK deployment files, if necessary.

3 Set the variable that defines the directory containing the DTK utilities (DT_TOOLS). This variable is set to %DT_PATH%\Tools by default in the sample script. Edit this path to reflect the actual location of the DTK utilities, if necessary.

4 Set the variable that defines the directory containing the configuration files for your Dell systems (DT_SYSTEMS). This variable is set to %DT_PATH%\Systems by default in the sample script.

5 Set the variable that defines the directory containing the DTK deployment scripts (DT_SCRIPTS). This variable is set to %DT_PATH%\Template\Scripts by default in the sample script. Edit this path to reflect the actual location of the DTK deployment scripts, if necessary.

6 Set the variable that indicates whether you want the deployment process to exit when any error is returned (DT_STOP_ON_ERROR). This variable is set to FALSE by default in the sample script, indicating that errors will be handed by the **ERRHNDL.BAT** script. Set the value to TRUE only if you want the deployment to exit on any error.

Specifying the SYSCFG Variables

The SYSCFG replication file is set to **syscfg.ini** by default. If you used the **SYSCAP.BAT** sample script to generate the **syscfg.ini** file, there should be no need to edit these default settings. If you have made any modifications to the sample scripts or to the suggested directory structure, you must confirm that **syscfg.ini** is correctly named and that the path is set correctly for your deployment.

To edit the SYSCFG capture variables, locate the following lines in **SYSCAP.BAT**:

```
set DT_SYS_CAPFILE=
%DT_SYSTEMS%\%DT_PLATFORM%\syscfg.ini
```

To edit the SYSCFG replication variables, locate the following lines in **SYSREP.BAT**:

```
set DT_SYS_REPFILE=
%DT_SYSTEMS%\%DT_PLATFORM%\syscfg.ini
```

NOTE: If the replication requires a change in the memory redundancy mode, the target server should be rebooted and the script should be executed again to complete the replication.

Specifying the RAC Configuration Variables

The IP address for your target system RAC is set in the sample script to `10.98.8.121` by default. Edit this value to reflect the actual IP address of your target system RAC, if applicable. Additionally, the RAC configuration file used to configure your RAC is set by default to **raccfg.ini**. If you use the **RACCAP.BAT** sample script to generate the **raccfg.ini** file, there should be no need to edit this default setting. If you have made any modifications to the sample scripts or to the suggested directory structure, you must confirm that **raccfg.ini** is correctly named and that the path is correctly set for your deployment. To edit the RAC configuration variables, locate the following lines in **RACCAP.BAT**:

```
set DT_RAC_CAPFILE=
%DT_SYSTEMS%\%DT_PLATFORM%\raccfg.ini
```

To edit the RAC configuration variables, locate the following lines in **RACREP.BAT**:

```
:: RAC section

set DT_NICCFG=10.98.8.121

set DT_RAC_CAPFILE=
%DT_SYSTEMS%\%DT_PLATFORM%\raccfg.ini
```

Specifying Variables Used to Create Disk Partitions and Install the Dell Utility Partition

NOTE: If the primary deployment state (`DT_PARTN`) is set to `OFF` in **PARTCFG.BAT**, the partitioning deployment state subsection is skipped.

The **PARTCFG.BAT** sample script provides the following disk partitioning default values. These values must be confirmed, in some cases, and modified, in other cases, before performing a deployment. Locate the **partn** label to find the following lines in **PARTCFG.BAT**:

```
set DT_PARTN_UP=ON

set DT_MOUNT=H


:: Default Hard Drive

set DT_HD=0
```

```
:: PARTCFG section
set DT_OS_SIZE=4000
set DT_OS_FSTYPE=FAT32

if not "%DT_PARTN_UP%"=="ON" goto ospart
:: DELLUP section
set DT_UP_SIZE=32
set DT_UP_IMAGE=%DT_SYSTEMS%\UPIMG.BIN

:: RAIDCFG section
set DT_VD_SIZE=10480
:: default size for virtual disk
```

CAUTION: `DT_MOUNT` **specifies the mount point of the Dell utility partition. If your RAMDRIVE, hard drive, or other device already owns this designation, mount will not override the setting and the utility partition will not be created.**

Perform the following steps to edit the default partition configuration default values, as necessary.

1. Edit the default utility partitioning variable (`DT_PARTN_UP=ON`), if necessary. This variable sets whether you want to install the Dell utility partition.

   NOTE: Dell recommends that you install the Dell utility partition.

2. Edit the default variable value for the default hard drive (`DT_HD=0`), if applicable.

   NOTE: Ensure that the default variable value for `DT_HD` is set to a valid non-removable disk, and not to removable media such as virtual media.

3. Edit the default variable value (in MB) for the primary operating system partition (`DT_OS_SIZE=4000`), if applicable.

4 Edit the default variable value for file system type (DT_OS_FSTYPE= FAT32), if applicable.

NOTE: The valid values are FAT32 and NTFS.

5 If the DT_PART_UP variable is set to ON, edit the default variable value (in MB) for the utility partition (DT_UP_SIZE=32), if applicable.

6 If the DT_PART_UP variable is set to ON, edit the default variable value for the utility partition installation package location (DT_UP_IMAGE=%DT_SYSTEMS%\UPIMG.BIN), if applicable.

You should have already copied the **UPIMG.BIN** file to the proper directory, as instructed in "Preparing the Directory Structure for Deployment in Microsoft Windows PE." Edit this variable only if you have changed the name or path of the utility partition installation package.

Editing DTK Task Scripts

The task scripts called by the master batch file do not require any edits (with the following exceptions) if the default variable values are utilized in the **TKENVSET.BAT** scripts. However, if you make any changes to the default values set by the **TKENVSET.BAT**, you must make the same edits to the matching default values set in the appropriate task scripts. Locate the "USER MODIFICATION REQUIRED" heading in the task scripts to find the variable values you might need to edit. Each of the following task scripts can be called by the user-written master batch script during a full deployment:

- **RACREP.BAT**
- **RAIDCFG.BAT**
- **SYSREP.BAT**
- **PARTCFG.BAT**

RACREP.BAT

The **RACREP.BAT** sample script applies RAC configuration options to the target system based on configuration information that has been saved in a configuration file. User input is optional, depending on variables set in the **RACREP.BAT** sample script. This script uses the **RACADM.EXE** utility to configure the RAC in the target system using the RAC configuration file, **raccfg.ini**, passed as the first parameter to this script. If this parameter is not

passed, the default variables values used in this script are set in the **TKENVSET.BAT** scripts. An optional second parameter can be passed to this file, which specifies the IP address to be used to configure the RAC. The **raccfg.ini** file is generated with the **RACCAP.BAT** sample script. See "RACCAP.BAT" for information about using the **RACCAP.BAT** sample script to capture RAC configuration information.

RAIDCFG.BAT

The **RAIDCFG.BAT** sample script configures RAID controllers detected on the system. User input is optional to run this script. The parameter is a name or path to a log file. This parameter is not set to any value by default, and no log file is generated. The default variable values used in this script are set in the **TKENVSET.BAT** scripts. This script uses the **RAIDCFG.EXE** utility to automatically configure the detected controllers on your target system. For example, for the first controller discovered, if **RAIDCFG.EXE** discovers only one attached hard drive, the script tries to create RAID 0; if two hard drives are discovered, a RAID 1 configuration is enabled; if three or more hard drives are discovered, a RAID 5 configuration is enabled.

NOTE: On the detected controller, a RAID virtual disk is created only if the array disks connected to that particular controller have space available and, where applicable, are not part of already existing array group.

SYSREP.BAT

The **SYSREP.BAT** sample script applies SYSCFG options to the target system based on configuration information that has been saved in a configuration file. User input is optional to run this script, depending on the variable settings in the **SYSREP.BAT scripts**. This script uses the **SYSCFG.EXE** utility to configure the BIOS and BMC in the target system using the configuration file, **syscfg.ini**, passed as the first parameter to this script. If this parameter is not passed, the default variables values used in this script are set in the **SYSREP.BAT** scripts. The **syscfg.ini** file is generated with the **SYSCAP.BAT** sample script. See "SYSCAP.BAT" for information about using the **SYSCAP.BAT** sample script to capture BIOS and BMC settings. The second parameter is a name or path to a log file. This parameter is not set to any value by default, and no log file is generated.

NOTE: If the replication requires a change in the memory redundancy mode, the target server should be rebooted and the script should be executed again to complete the replication.

PARTCFG.BAT

The **PARTCFG.BAT** sample script creates and populates the Dell utility partition using **UPINIT.BAT** and creates the operating system partition on a specified disk. (See "UPINIT.BAT" for more information about using the **UPINIT.BAT** tool.) User input is not required to run this script. The default variable values used in this script are set in the **TKENVSET.BAT** scripts.

NOTE: Ensure that `DT_HD` is set to the required disk to be configured before you execute the **PARTCFG.BAT** script.

Using the DTK Sample Scripts to Capture Configuration Information in Windows PE

The **SYSCFG.EXE** and **RACADM.EXE** utilities can read the source system configuration and duplicate that configuration on a target system using a sample DTK script file. These configuration files are used during a full deployment to configure the target system's BIOS, BMC, and RAC settings. The following sections provide instructions for using the **SYSCAP.BAT** and **RACCAP.BAT** sample scripts to create configuration files.

Capturing the System BIOS and BMC Configuration With the SYSCAP.BAT Sample Script

You can replicate the system BIOS and BMC configurations from a source Dell system to an identical target Dell system (for example, from one PowerEdge™ 1850 to another PowerEdge 1850).

NOTE: For the script to complete successfully, the source system must have access to the network share that includes the prebuilt DTK directory structure.

SYSCAP.BAT

The **SYSCAP.BAT** sample script captures BIOS and BMC configuration settings and saves them to a specified location. An optional user input can be passed as a parameter. You can provide the path or filename as an input to this script. The default variable values used in this script are set in the script to the correct default values to perform a scripted deployment. The **SYSCAP.BAT** script uses the **SYSCFG.EXE** utility to retrieve BIOS and BMC configuration settings from the source system and to save the settings in a file named **syscfg.ini** in the `\Toolkit\Systems\<target system>` directory.

Perform the following steps to run the **SYSCAP.BAT** sample script to capture BIOS and BMC configuration information in the **syscfg.ini** file:

1. Edit the **startnet.cmd** file on your Windows PE image to call the **SYSCAP.BAT** script after the network share with the prebuilt DTK directory structure is mounted. For example, add:

   ```
   call
   Z:\Dell\x32\Toolkit\Template\Scripts\SYSCAP.BAT
   ```

2. Boot the source system into Windows PE.

   The script runs and saves configuration information in the `\Toolkit\Systems\<target system>` directory.

3. Edit the **startnet.cmd** file in your Windows PE image to remove the text you added in step 1.

Capturing the RAC Configuration With the RACCAP.BAT Sample Script

If your system has a RAC, you can replicate the RAC configuration from a source Dell system's RAC to an identical target Dell system with an identical RAC (for example, from one PowerEdge 1850 with a DRAC4/I to another PowerEdge 1850 with a DRAC4/I).

NOTE: For the script to complete successfully, the source system must have access to the network share that includes the prebuilt DTK directory structure.

RACCAP.BAT

The **RACCAP.BAT** sample script captures RAC configuration settings and saves them to a specified location. An optional user input can be passed as a parameter. You can provide the path or file name as an input to this script. The default variable values used in this script are set in the script to the correct default values to perform a scripted deployment. The **RACCAP.BAT** script uses the **RACADM.EXE** utility to retrieve RAC configuration settings from the source system and to save the settings in a file named **raccfg.ini** in the **\Toolkit\Systems\<target system>** directory.

NOTE: Ensure that you run **RACCAP.BAT** in a writable environment. If you run **RACCAP.BAT** in a read-only environment, DTK displays the message `<filename.ini> has been generated` even though no new file is created.

Perform the following steps to run the **RACCAP.BAT** sample script to capture RAC configuration information in the **raccfg.ini** file:

1. Edit the **startnet.cmd** file on your Windows PE image to call the **RACCAP.BAT** script after the network share with the prebuilt DTK directory structure is mounted. For example, add:

   ```
   call
   Z:\Dell\x32\Toolkit\Template\Scripts\RACCAP.BAT
   \Z:\raccfg.ini
   ```

2. Boot the source system into Windows PE.

   The script runs and saves configuration information in the **\Toolkit\Systems\<target system>** directory.

3. Edit the **startnet.cmd** file in your Windows PE image to remove the text you added in step 1.

NOTE: The RAC configuration file is referred to as the **.cfg** file in the *Dell Remote Access Controller Racadm User's Guide and the Dell Remote Access Controller 4 User's Guide*. See the *Dell Remote Access Controller Racadm User's Guide or the Dell Remote Access Controller 4 User's Guide* for additional information about manually creating a RAC configuration file.

NOTE: Do not replicate the RAC IP address when creating a **.cfg** file. Replicating the RAC IP address can leave the system inaccessible because multiple systems are configured with the same IP address.

UPINIT.BAT

UPINIT.BAT is a tool used to create and populate the Dell utility partition.

NOTE: You can use this script to create a new Dell utility partition only if there are no existing partitions on your hard disk.

You can also use this script to upgrade or downgrade an existing Dell utility partition. Upgrades or downgrades can be made only to existing Dell utility partitions using the overwrite option.

CAUTION: The existing Dell utility partition should be the first primary partition on the hard disk. Also, the Dell utility partition must be greater than or equal to 32 MB and less than 2 GB.

For details on the command line arguments to be used for **UPINIT.BAT**, refer to the *Deployment Toolkit Command Line Interface Reference Guide*.

Preparing the Operating System Installation Scripts for Windows PE

The **WININST.BAT** and **W2K8INST.BAT** scripts are used to install supported Windows operating systems.

WININST.BAT

The **WININST.BAT** sample script is used to perform an unattended installation of Windows 2000 Server and Windows Server 2003 operating systems. The variables and paths must be specified before the operating system can be installed.

CAUTION: It is strongly recommended that you consult your Windows operating system documentation and unattended deployment documentation to develop a thorough understanding of the Windows unattended installation process before attempting to perform a full scripted deployment.

The syntax for **WININST.BAT** is

```
wininst.bat [W2K|NET]
```

Choose `W2K` (default) to install Windows 2000 Server or `NET` to install Windows Server 2003.

Utilities Used

WINNT32.EXE — Used to install the operating system in conjunction with the **UNATTEND.TXT** answer file.

External Dependencies

The DT_PLATFORM variable is set by **TKENVSET.BAT** using platform discovery.

The **WININST.BAT** script uses the **UNATTEND.TXT** file in conjunction with **WINNT32.EXE** utility to install the operating system files. See "UNATTEND.TXT" for more information about preparing the **UNATTEND.TXT** file.

Specifying the Windows Installation Variables and Installation Paths

To edit the Windows operating system installation variables, locate the following lines in **WININST.BAT**:

```
:: The location where source files for the OS reside

set DT_OSSRC=%DT_DRIVE%\W2KSRC\I386


:: Location of Unattend.txt file

set DT_UNATTEND=
%DT_SYSTEMS%\%DT_PLATFORM%\unattend.txt


:: Location of platform-specific Dell drivers

set DT_OEM=%DT_SYSTEMS%\%DT_PLATFORM%\configsets
```

After you have either confirmed or edited the Windows operating system installation variables, you are ready to prepare the **UNATTEND.TXT** file to be used in the operating system installation.

UNATTEND.TXT

The **UNATTEND.TXT** file is the default name of the answer file that you use to automate Windows Setup during an unattended installation. A sample copy of the **UNATTEND.TXT** file is provided as part of the DTK download in the **\Toolkit\Template\Configs** directory. **UNATTEND.TXT** contains the headings and parameters that instruct Windows Setup to perform various configuration tasks. Before a full deployment can be run, you must edit all appropriate headings and parameters, as necessary.

NOTE: See your operating system documentation for instructions on modifying the options in the operating system unattended installation file to customize the scripted installation of your operating system.

When you have completed all necessary modifications of the **UNATTEND.TXT** file, save the completed file in the **\Toolkit\Systems\<target system>** directory.

W2K8INST.BAT

The **W2K8INST.BAT** sample script is used to perform an unattended installation of Windows Server 2008 operating system. The variables and paths must be specified before the operating system can be installed.

CAUTION: It is strongly recommended that you consult your Windows operating system documentation and unattended deployment documentation to develop a thorough understanding of the Windows unattended installation process before attempting to perform a full scripted deployment.

The syntax for **W2K8INST.BAT** is

```
w2k8inst.bat
```

Utilities Used

SETUP.EXE — Used to install the operating system in conjunction with the **UNATTEND.XML** answer file.

External Dependencies

W2K8INST.BAT uses two environment variables: DT_PLATFORM and DT_DRIVE.

The DT_PLATFORM variable is set by **TKENVSET.BAT** using platform discovery.

The DT_DRIVE denotes the network drive letter to the network shared path containing the Windows Server 2008 operating system source and the **UNATTEND.XML** file.

The **W2K8INST.BAT** script uses the **UNATTEND.XML** file in conjunction with **SETUP.EXE** utility to install the operating system files. See "UNATTEND.XML" for more information about preparing the **UNATTEND.XML** file.

Specifying the Windows Installation Variables and Installation Paths

To edit the Windows operating system installation variables, locate the following lines in **W2K8INST.BAT**:

```
:: The location where source files for the OS reside

set DT_OSSRC=%DT_DRIVE%\sources
```

```
:: Location of Unattend.xml file

set DT_UNATTEND=
%DT_SYSTEMS%\%DT_PLATFORM%\unattend.xml


:: Location of platform-specific Dell drivers

set DT_OEM=
%DT_SYSTEMS%\%DT_PLATFORM%\$OEM$%DT_WINDOWS%
```

UNATTEND.XML

The **UNATTEND.XML** file is the default name of the answer file that you use to automate Windows Setup during an unattended installation. A sample copy of the **UNATTEND.XML** file is provided as part of the DTK download in the **\Toolkit\Template\Configs** directory.

UNATTEND.XML contains the headings and parameters that instruct Windows Setup to perform various configuration tasks. In **UNATTEND.XML**, you must specify various setup options, including how to partition disks and the location of the Windows image you want to install. Before a full deployment can be run, you must edit all appropriate headings and parameters, as necessary.

NOTE: See your operating system documentation for instructions on modifying the options in the operating system unattended installation file to customize the scripted installation of your operating system.

When you have completed all necessary modifications of the **UNATTEND.XML** file, save the completed file in the **\Toolkit\W2K8_deploy\<target system>** directory.

After you have successfully edited your deployment scripts, you are ready to create your deployment media and run your deployment. For instructions on creating deployment media and running a full deployment, see "Running Deployment Scripts Using DTK and Microsoft Windows PE."

5

Running Deployment Scripts Using DTK and Microsoft Windows PE

This section provides best practices procedures and scenarios for using the Deployment Toolkit (DTK) to perform pre-operating system configuration tasks and to install supported operating systems on supported Dell™ systems.

After you have populated the deployment directory structure with all necessary files and carefully edited the deployment scripts and configurations files, you are ready to begin the final stage of the deployment process. Before running the deployment, however, you need to create specific deployment media to facilitate your chosen deployment method. The common deployment scenarios are:

- CD-based local deployment
- CD-based local deployment with networking enabled
- Using a third-party deployment solution framework

For the CD-based deployment methods, you begin by creating and customizing Microsoft® Windows® PE for Dell systems.

Creating and Customizing Images for Windows PE 1.0

Obtaining Windows OEM Preinstallation Kit (OPK) or Windows PE CD

Contact Microsoft to obtain the latest version of OPK or Windows PE CD.

Copying Tools From OPK or Windows PE CD on a Development System

1. Create and name a directory on your development system hard drive, such as **\WinPE_OPKTools**.
2. Insert the Windows OPK CD or Windows PE CD into your development system CD drive.
3. Copy all of the CD files from the Windows PE directory (**<cd-drive>:\winpe*.***) to the **\WinPE_OPKTools** directory.

4 Copy **factory.exe** and **netcfg.exe** from **<cd-drive>:\tools\x86** directory to the **\WinPE_OPKTools** directory.

5 Remove the Windows OPK CD or Windows PE CD from the CD drive.

Creating a Standard Windows PE

For building a standard Windows PE with OPK tools, refer to the help files available on the OPK CD. You can get to the help files after you install OPK on your development system. Brief instructions are provided here on how to build a standard Windows PE.

NOTE: You must have read-write permissions on your development system. You cannot run **mkimg.cmd** from a read-only device, such as a CD drive.

1 Create and name a directory on your development system hard drive such as **\DellWinPEBuild**. This directory serves as a **<destination_directory>** used with **mkimg.cmd** to create Windows PE.

2 Insert the 32-bit Windows Server® 2003 (with SP1) product CD into the CD drive.

3 Open a command prompt and navigate to the `\WinPE_OPKTools` directory you created earlier.

4 Run the command: `mkimg.cmd <source_directory> <DellWinPEBuild> /PNP /WMI.`

NOTE: The `/PNP` option enables support for Plug and Play device drivers; the `/WMI` option enables support for Windows Management Instrumentation.

NOTE: Ensure that you include the `/WMI` option with **mkimg.cmd** for the DTK tools to run properly.

The **<source_directory>** specifies the location of the Windows Server 2003 product CD.

- If the **<source_directory>** is a CD drive, you need to specify only the drive letter.
- If the **<source_directory>** is a network path, the path must be to the parent directory of the `\I386` directory.

NOTE: Do not include a trailing slash in the **<source_directory>**.

NOTE: If you plan to use virtual media within your Windows PE environment, ensure that your Windows PE image has **cdrom.sys** and **flpydisk.sys** drivers.

The **\DellWinPEBuild** specifies the path where the files necessary to create the Windows PE are temporarily stored. If this directory does not already exist, **mkimg.cmd** creates it.

This process extracts all the necessary files from **\I386** on the Windows Server 2003 product CD and puts them in **\DellWinPEBuild** drive. These files include drivers that are part of the Windows Server 2003 CD. These drivers may be old and may not recognize the new peripherals integrated in your new Dell system. Proceed to add the latest drivers in the standard Windows PE build you just created.

Integration of DTK Directory Structure in Windows PE Build

This task involves a five-step process:

1. Integrating DTK tools and scripts: Copy the folder **\dell** that you unzipped from the Dell-supplied zip file to **<DellWinPEBuild>,** or extract the Dell-supplied file directly into the Windows PE build.
2. Navigate to the **\Dellwinpebuild\Dell\x32\Drivers\winpe1.x** location and run the following command:

   ```
   driverinst.bat <path to Dellwinpebuild> <path to
   dell\drivers>
   ```

3. Adding support for mass storage drivers into Windows PE:

   NOTE: You must implement this step to ensure mass storage drivers are installed into Windows PE.

   In the **winpeoem.sif** file (available under **I386\SYSTEM32** in your Windows PE directory), edit the following text to remove the semicolons from the `Oem Driver Params` section and append the directory names for the Dell mass storage drivers:

   ```
   OemDriverRoot=""
   OemDriverDirs=MRAID, PERC4IM, PERC5, SAS5
   ```

   NOTE: A sample winpeoem.sif file is available under **\Dell\Toolkit\template\Configs**.

   NOTE: When booting the target system, ensure that the RAC and the mr2kserv services are running.

4 Extracting the following files from your Windows Server 2003 product CD to **\Dell\Toolkit\Tools**:

- rpcns4.dl_
- rpcrt4.dl_
- rpcss.dl_

NOTE: The directory name should not have any spaces.

NOTE: You can now proceed to customize Windows PE per your requirements.

Creating a Bootable Windows PE ISO Image

After you have completed the DTK integration and customization, you are now ready to convert all the files in **<DellWinPEBuild>** into an ISO image using the OPK tool. Follow the steps below to create a bootable ISO image from **<DellWinPEBuild>**.

1 Create and name a directory on your development system hard drive such as **FINALBUILD**. This is where the ISO file is stored.

2 Open a command prompt and navigate to the **\WinPE_OPKTools** directory you created earlier.

3 Run the command: `OSCDIMG -bETFSBOOT.COM -n <DellWinPEBuild> <iso_filename>`

Here, `ETFSBOOT.COM` (boot sector file) is a part of the OPK tools you copied in **\WinPE_OPKTools** directory. These files make the Windows PE CD bootable. For example, if the final target folder for storing ISO image is **C:\FINALBUILD**, and your Windows PE file set is in **<DellWinPEBuild>**, the following command line creates a bootable ISO image called **Dellx86winpe.iso**:

```
C:\WINPE_OPKTOOLS\OSCDIMG -bETFSBOOT.COM -n
c:\DELLWINPEBUILD

C:\FINALBUILD\Dellx86winpe.iso
```

After you create an ISO image, you can use any CD burning software to burn the image onto a CD. After burning the ISO image, ensure that it boots from the CD drive for all the supported Dell systems you plan to deploy. After it boots, you are advised to test all the tools and scripts on these systems to ensure that the integration went well and there are no issues with hardware components not being recognized.

CD-Based Local Deployment

You are now ready to use your bootable CD to call the deployment components from the directory structure on the CD:

1. Boot the target system with the bootable deployment CD.
2. Execute the master batch file, which calls individual task scripts and utilities from the CD to complete the deployment process.

Running the Windows PE 1.0 Image

CD-Based Local Deployment With Networking Enabled

You can also use your bootable CD to call the deployment components from the directory structure on the CD and the remaining components from the network share:

1. Boot the target system with the bootable deployment CD.
2. Execute the master batch file, which calls individual task scripts and utilities from the network share to complete the deployment process.

Using a Third-Party Deployment Solution Framework

You can use the DTK with any existing third-party deployment solution framework that provides a Automated Deployment Services (ADS) booting infrastructure that can be used as the transport mechanism for the DTK utilities. Because each third-party deployment framework is unique, however, these solutions fall outside the scope of this document. If you plan to utilize a third-party deployment solution framework, keep in mind that the DTK is a Windows PE-based set of tools and scripts, so the deployment solution framework must also support Windows PE as a pre-operating system environment.

Creating and Customizing Images for Windows PE 2.0

If you are using Windows PE 2.0, download Windows Automated Installation Kit (Windows AIK) from the Microsoft website at **www.microsoft.com**. By default, Windows AIK is copied to the **C:\Program Files\Windows AIK** directory.

NOTE: DTK 3.2 for 64-bit systems supports only Windows PE 2.0.

Integration of DTK Directory Structure in Windows PE Build

DTK provides a script, **VPE_driverinst.bat**, to pre-install the Dell drivers into a base Windows PE 2.0 image (**winpe.wim**). Use the following steps to execute this script in 32-bit or 64-bit supported systems:

1. Open a command prompt on your system and change the directory to the location of **VPE_driverinst.bat**.

   Example for 64-bit supported systems:

   ```
   cd C:\Dell\x64\Drivers\winpe2.x
   ```

   Example for 32-bit supported systems:

   ```
   cd C:\Dell\x32\Drivers\winpe2.x
   ```

2. Execute **VPE_driverinst.bat** with two required arguments: **<WINPEPATH>** and **<DTKPATH>**. For example:

   ```
   VPE_driverinst.bat <WINPEPATH> <DTKPATH>
   ```

   Where **<WINPEPATH>** is the destination path to create the directory structure for Windows PE 2.0 and **<DTKPATH>** is the path to Dell drivers in the extracted DTK toolkit. For example:

   ```
   VPE_driverinst.bat C:\vistaPE_x86 C:\DELL\DRIVERS
   ```

   This will pre-install the Dell drivers into **winpe.wim**.

   NOTE: *<WINPEPATH>* is passed as the destination to the Windows AIK command `copype.cmd`. The destination folder **C:\vistaPE_x86** is created as part of the process, and should not already exist.

3. Replace the customized **winpe.wim** image to the folder **VistaPE_x86\ISO\sources**. For example:

   ```
   Copy c:\vistaPE_x86\winpe.wim
   c:\vistaPE_X86\ISO\sources\boot.wim
   ```

Creating a Bootable CD

Use the following steps to create bootable media:

1. Click **Start**, navigate to **All Programs→Microsoft Windows AIK**.
2. Click **Windows PE Tools Command Prompt** to open a command prompt window.
3. Run the following command:

```
oscdimg -n -bc:\vistaPE_x86\etfsboot.com
c:\vistaPE_x86\ISO c:\vistaPE_x86\WinPE2.0.iso
```

This command creates a CD bootable ISO image called **WinPE2.0.iso**.

You can use any CD burning software to burn the image onto a CD.
After burning the ISO image, ensure that it boots from the CD drive for all the supported Dell systems you plan to deploy. After it boots, you are advised to test all the tools and scripts on these systems to ensure that the integration went well and there are no issues with hardware components not being recognized.

Running the Windows PE 2.0 Image

CD-Based Local Deployment

You are now ready to use your bootable CD to call the deployment components from the directory structure on the CD:

1. Boot the target system with the bootable deployment CD.
2. Execute the master batch file, which calls individual task scripts and utilities from the CD to complete the deployment process.

CD-Based Local Deployment With Networking Enabled

You can also use your bootable CD to call the deployment components from the directory structure on the CD and the remaining components from the network share:

1. Boot the target system with the bootable deployment CD.
2. Execute the master batch file, which calls individual task scripts and utilities from the network share to complete the deployment process.

Using a Third-Party Deployment Solution Framework

You can use the DTK with any existing third-party deployment solution framework that provides a Automated Deployment Services (ADS) booting infrastructure that can be used as the transport mechanism for the DTK utilities. Because each third-party deployment framework is unique, however, these solutions fall outside the scope of this document. If you plan to utilize a third-party deployment solution framework, keep in mind that the DTK is a Windows PE-based set of tools and scripts, so the deployment solution framework must also support Windows PE as a pre-operating system environment.

6

Choosing an Embedded Linux-Based Deployment Method

A well-thought-out deployment plan is critical to the success of your deployment effort and includes the following considerations:

- Assessing your existing IT environment
- Selecting an operating system
- Selecting the optimum Dell™ OpenManage™ systems management software configuration for your system
- Choosing a deployment method

The Deployment Toolkit (DTK) is designed to be flexible enough to fit into almost any deployment plan. For the purposes of this document, however, the focus is on two specific deployment methods: CD-based deployment (local) and network-based deployment (remote).

Choosing a Deployment Method for Embedded Linux

The DTK components can be used in many different ways to assist IT staff in deploying Dell systems in large numbers. Because all the tools and scripts are task oriented, many of them can be used separately to perform a specific task on many systems or collectively to perform many tasks on many systems at once.

The configuration complexity and pros and cons of each deployment method are provided in Table 6-1. After you have decided which deployment method best fits your needs, you are ready to begin preparing for the deployment.

Network-Based Deployment

NOTE: The DTK network-based deployment is not supported from mapped Novell® NetWare® systems.

The general network-based deployment process includes the following steps:

1. Modify the appropriate section in **pxe config** file.
2. Boot the target server from PXE.

   The DTK start-up phase mounts the network share specified in the **pxe config** file.

   The DTK start-up phase runs the script (name specified in the **pxe config** file) from the network.

Alternatively, you can integrate the DTK components into an existing third-party deployment solution framework. See "Using a Third-Party Deployment Solution Framework."

CD-Based Deployment With Network Access

The general steps include:

1. Copy the DTK ISO image into a share.
2. Modify the appropriate section in the **isolinux/isolinux.cfg** file.
3. Reburn the contents into a CD (this will be the master CD).
4. Boot the target server from the master CD.

   The DTK start-up phase mounts the network share specified in the **isolinux.cfg** file.

   The DTK start-up phase runs the script (name specified in the **isolinux.cfg** file) from the network.

Creating a Bootable CD With Network Stack

You will need a bootable CD to use in a remote deployment. This bootable CD must have the embedded Linux network stack with specific network drivers to facilitate networking. After you have a network share created and populated with all the necessary DTK components and supporting files, you can access all required files by mounting a network share from the target system. You can perform all of these tasks automatically. This deployment method is recommended if a network connection is available to the target

system—the space on a network share can be very large, and a single set of utilities and scripts can be accessed by many systems, making the deployment process easy to manage.

After a network drive is mapped from a target system to a network share, all of the utilities and scripts are available to use. The master deployment script on the network share can be called from the bootable CD to perform a task or multiple tasks on the target server. For example, pre-operating system configuration tasks can be deployed from the network and operating system files can be installed.

Creating a Customized Bootable CD with Network Stack

See "Running Deployment Scripts Using DTK and Embedded Linux" for details on how to create a customized version of a single embedded Linux bootable CD that works across all Dell systems and provides an underlying network stack.

Using a Third-Party Deployment Solution Framework

You can use the DTK with any existing third-party deployment solution framework that provides a PXE booting infrastructure that can be used as the transport mechanism for the DTK utilities. Because each third-party deployment framework is unique, however, these solutions fall outside the scope of this document. If you plan to utilize a third-party deployment solution framework, keep in mind that the DTK is an embedded Linux-based set of tools and scripts, so the deployment solution framework must also support embedded Linux as a pre-operating system environment.

CD-Based Local Deployment for Embedded Linux

The general CD-based deployment process includes the following steps:

1. Copy the DTK ISO image into a share.
2. Modify the appropriate section in the **isolinux/isolinux.cfg** file.
3. Reburn the contents into a CD (which will be the master CD).
4. Boot the target server from the master CD.

   The DTK startup phase mounts the CD.

   The DTK startup phase runs the script (name specified in the **isolinux.cfg** file) from the CD.

This deployment method is the easiest to assemble, but is the least flexible. Any change to the system configuration, for example, requires the creation of a new bootable CD. This method does not provide a writable media making it necessary to use predefined and tested configuration files.

Creating a Bootable CD

Bootable CDs can be created using any commonly available CD burning software in a two-part process that requires a self-bootable ISO image with drivers that load from system files. When you create a bootable CD, this image is supplied to the CD burning software along with the complete deployment directory structure, which includes all necessary supporting files to perform the deployment. The CD burning software creates an image of the self-bootable ISO image and the deployment files on the CD.

NOTE: See "Running Deployment Scripts Using DTK and Embedded Linux" for more information about creating an embedded Linux-bootable CD.

Using DTK Utilities and Scripts From a Bootable CD

The bootable CD can be configured to start and execute the master deployment file during the booting process. The master deployment file performs all necessary pre-operating system configuration tasks and then installs the operating system from the files in the deployment directory structure on the CD.

NOTE: You must write the master deployment file. This file is not provided with the sample scripts.

NOTE: You must create a RAMDISK for temporary data storage when using a bootable CD for a local deployment.

Creating a Customized Bootable CD

See "Running Deployment Scripts Using DTK and Embedded Linux" for details on how to create a customized version of a single embedded Linux image that can work across all Dell systems.

Using DTK Utilities and Scripts From a Customized Bootable CD

After the customized bootable embedded Linux CD is created, IT administrators can take advantage of the sample scripts to automate the entire deployment process. In this purely CD-based deployment, it is necessary to add the operating system installation files on the CD media.

Deployment Matrix (Embedded Linux)

Depending on the scope, current business needs, network setup, and process, you might select any of the suggested methods of deployment. Table 6-1 outlines four typical methods of deployment and the environment most suitable for each.

Table 6-1. Deployment Method Matrix

Deployment Methods	Complexity	Advantages	Disadvantages
Local deployment (bootable CD)	Relatively simple	• Necessary if a network connection is not available or is too slow. • Can perform a minimal installation of supported Linux operating systems.	• Any change requires the creation of a new bootable CD. • Limited space on CD for some operating system installation files. • Deployment tasks must be performed at the individual target system. • CDs are read-only media and require the creation of a RAMDISK for temporary data storage. • Installation time is high.

Table 6-1. Deployment Method Matrix *(continued)*

Deployment Methods	Complexity	Advantages	Disadvantages
Remote deployment over a network (bootable embedded Linux CD with network stack loaded)	Moderate	• Everything is in one place in a network share. • Easy to manage—changes have to be made in only one place. • Data captured (profiles) can be stored in a network share. • Data can be replicated from a network share. • Can install supported Linux operating systems.	• Must have network connection. • Deployment tasks must be performed at the individual target system. • Must locate the appropriate embedded Linux network drivers. • Not supported from a mapped NetWare system.
Deployment solution framework from a third-party vendor	Mixed	• The DTK can be used in the context of third-party deployment solution framework. • Third-party deployment solution framework is used as the deployment transport mechanism. • Tasks and scripts can be pushed to the target systems.	Must either acquire and learn to use or have an existing third-party deployment solution framework.

Table 6-1. Deployment Method Matrix *(continued)*

Deployment Methods	Complexity	Advantages	Disadvantages
PXE boot	High	• Everything is in one place on a network share. • Easy to manage—changes have to be made in only in one place. • High flexibility for remote deployment. • No media required. • Faster deployment.	• Must have high-speed connectivity to network (LAN). • Set up time is longer.

After you have determined which deployment method best suits the needs of your organization, you are ready to begin building the DTK deployment directory structure on a network volume. For instructions on installing the DTK components and populating the deployment directory structure, see "Preparing the Directory Structure for Deployment in Embedded Linux."

7

Preparing the Directory Structure for Deployment in Embedded Linux

This section describes some of the procedures necessary to begin the deployment process, including:

- Downloading and unzipping the Deployment Toolkit (DTK) components, including a comprehensive matrix of the DTK components
- Setting up a directory structure to facilitate a successful deployment, including a list of the supporting components that are not included with the DTK, but are needed to use the DTK
- Setting up an optimally configured source system

Upgrading Your DTK Version

When upgrading from a previous release of the DTK 2.*xx* version, ensure that you create the new DTK version 3.*xx* environment separately from your previous version of DTK. This step is necessary because the environments and requirements for DTK version 3.*xx* are completely different.

Installation for Embedded Linux

The DTK components are provided in an ISO image on the Dell™ Support website at **support.dell.com**. You can use a CD-burning software to burn the ISO image on a bootable CD or loop mount the ISO image to access the image contents.

Perform the following steps to extract the DTK components to a workstation running a Linux operating system:

1. Download the ISO image **dtk_3.X_XXX_Linux.iso** from the Dell Support website at **support.dell.com** and save it on your system (where X is the latest version of DTK and *xxxx* is the latest build for the DTK release).
2. After the image downloads, burn it on a CD.

   The directory structure on the CD (after it is mounted) is shown in Figure 7-1.

Figure 7-1. CD Directory Structure

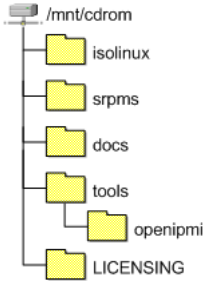


Table 7-1 describes the files, components, and other contents of the mounted CD.

Table 7-1. Mounted CD Contents

Contents	Details	Location
dchipm32.ini	Configuration file for the instrumentation software.	**/opt/dell/toolkit/lib/bin**
docs	Contains: • *Command Line Interface Reference Guide* • *User's Guide*	**/mnt/cdrom/docs**
LICENSING	Contains all the licenses for the open source components used in this CD.	**/mnt/cdrom/ LICENSING**
omreg.cfg	Configuration file for the instrumentation software. **NOTE:** DTK utilities will fail if this configuration file is not present in `/etc`.	**/mnt/cdrom/tools**
openipmi	Contains IPMI drivers and support scripts to install them. These drivers are required for all Baseboard Management Controller (BMC) and Dell Remote Access Controller (DRAC) related commands.	**/mnt/cdrom/tools**

Table 7-1. Mounted CD Contents *(continued)*

Contents	Details	Location
prereqcheck	Contains files that are used by **srvadmin-openipmi.sh** to run preliminary requisite checks.	**/mnt/cdrom/tools/openipmi/supportscripts**
sa.1	Contains the Linux kernel images	**/mnt/cdrom/isolinux**
sa.2	Initial RAM disk containing the embedded Linux component and the DTK tools	**/mnt/cdrom/isolinux**
Source RPMS (SRPMs)	Contains open source components used in the Dell-provided embedded Linux	**/mnt/cdrom/srpms**
srvadmin-hapi.<version number>.i386.rpm	Contains a set of drivers and libraries used by the DTK tools to interact with the hardware. Used to build your custom ISO.	**/mnt/cdrom/tools**
srvadmin-omilcore-hapi.<version number>.i386.rpm	Contains a set of drivers and libraries used by the DTK tools to interact with the hardware. Used to build your custom ISO.	**/mnt/cdrom/tools**
srvadmin-openipmi.sh	Checks if the OpenIPMI driver needs to be installed or upgraded, and if required, attempts to do so. Also displays status information and information about any action it attempts.	**/mnt/cdrom/tools/openipmi/supportscripts**
start-stage3.sh	The startup script used by the toolkit ISO image to boot.	**/mnt/cdrom/tools**
start-hapi.sh	The startup script to load and configure the HAPI drivers.	**/mnt/cdrom/tools**
start-raid.sh	Contains the basic setup for **raidcfg** to work.	**/mnt/cdrom/tools**

Boot the CD onto a target system. Figure 7-2 shows the directory structure on the target system.

Figure 7-2. Target System Directory Structure

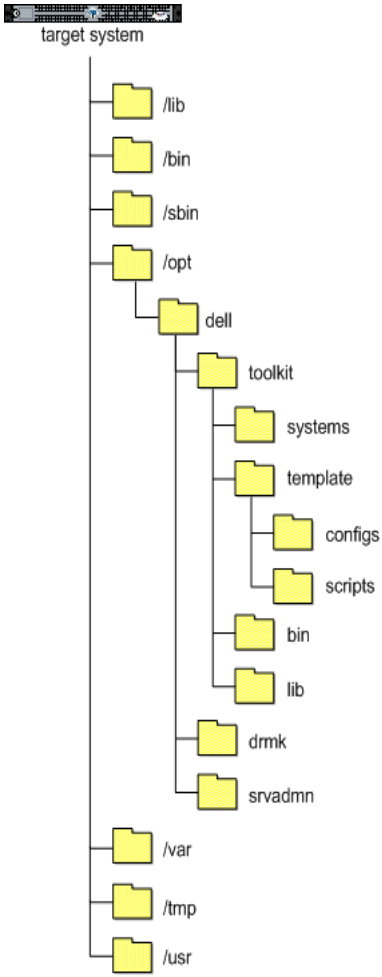


Table 7-2 describes the contents of the directory structure on the target system.

Table 7-2. Target System Directory Structure Contents

Contents	Details
/lib	The libraries and drivers required for embedded Linux to function.
/bin	Basic utilities (ls, chmod, and so on)
/sbin	Utilities that are required by the system superuser to perform administrative tasks.
/opt	The libraries, binaries, and scripts required for the toolkit to function, which are available at: `/mnt/cdrom/tools/dell.tar.gz`. **NOTE:** The **/opt/dell/srvadmin** directory should be writable.
/var	Any run-time variable data that might be required for embedded Linux and tools to function.
/tmp	The only safe, writable area. However, you will lose all data in this folder every time you reboot.
/usr	Miscellaneous tools and libraries for embedded Linux to function.

DTK Contents

The DTK contains embedded Linux-based utilities, sample scripts, sample configuration files, drivers, and documentation for automating the deployment of Dell systems. The following sections provide a complete list of the DTK contents and a short description of each component. For a more complete explanation of the DTK utilities, see the *Command Line Interface Reference Guide*. For an explanation of the sample configuration files and scripts, see "Preparing the Script Files for Deployment Using Embedded Linux."

System Utilities and Supporting Files

Table 7-3 describes the files, located at **/opt/dell/toolkit/bin** and the supporting libraries located in **/opt/dell/toolkit/lib**. See the *Command Line Interface Reference Guide* for complete information about the capabilities of each DTK utility.

Table 7-3. System Utilities and Supporting Files

Toolkit Components	Purpose
config.ini	Configuration file used by **raidcfg**.
pci.ids	Used with **syscfg** to report PCI devices.
racadm	Wrapper script which invokes **racadm3** or **racadm4**, depending on the remote access controller (RAC) on your system.
racadm3	Configures Dell Remote Access Controller (DRAC) III, DRAC III/XT, embedded remote access (ERA), ERA/O, and ERA/MC RACs.
racadm4	Configures DRAC 4/P and DRAC 4/I RACs.
racadm5	Configures DRAC 5 RACs.
racadm6	Configures iDRAC6 **NOTE:** This feature may not be supported on all systems.
racadmErrorCodes.txt	Lists the error codes and messages for the **racadm** tool.
racser	Chat script to bring up a Point-to-Point Protocol (PPP) connection required to transport **racadm3**.
racsvc3.sh	Start-up script for DRAC III, DRAC III/XT, ERA, ERA/O, and ERA/MC RACs.
racsvc4.sh	Start-up script for DRAC 4/P and DRAC 4/I RACs.
raidcfg	Creates and reports RAID information.
RaidcfgErrorCodes.txt	Lists the error codes and messages for the **raidcfg** tool.
stsvc.ini	Configuration file used by **raidcfg**.
syscfg	• Reports and configures: – basic input/output system (BIOS) settings and options – baseboard management controller (BMC) settings and options • Maintains system state between reboots • Reports hardware devices and other system-related information
SyscfgErrorCodes.txt	Lists the error codes and messages for the **syscfg** tool.

Table 7-3. System Utilities and Supporting Files *(continued)*

Toolkit Components	Purpose
sys.ini	A **.ini** file used with **syscfg.**
UpinitErrorCodes.txt	Lists the error codes and messages for **upinit.sh**.

Sample Configuration Files

Table 7-4 describes examples of configuration files used to perform a scripted deployment. These files are located at **opt/dell/toolkit/template/configs**.

NOTE: All configuration files are provided as examples only. These sample configuration files must be edited before they can be used in any deployment tasks.

Table 7-4. Sample Configuration Files

Toolkit Components	Purpose
autoinst.xml	Sample file used with unattended installation of SUSE® Linux Enterprise Server operating systems.
ks.cfg	Sample file used with unattended installation of Red Hat® Enterprise Linux® Server (versions 3 and 4) operating systems.
ks-rhel5.cfg	Sample file used with unattended installation of Red Hat Enterprise Linux Server (version 5) operating system.
raccfg.ini	Sample configuration file for DRAC 4 RACs.
raidcfg.ini	Sample output file to enable RAID replication.
syscfg.ini	Sample configuration file for SYSCFG utility to configure the BIOS and BMC settings.

Sample Scripts

Table 7-5 lists sample scripts necessary to perform a scripted deployment.

NOTE: All DTK scripts are provided as examples only. The batch scripts and configuration files must be modified to reflect the unique information for each deployed system.

Table 7-5. Sample Scripts

Toolkit Components	Purpose
errhndl.sh	Handles errors returned in the DTK scripts.
lininst.sh	Performs an unattended installation of a supported Red Hat Enterprise Linux Server operating system.
partcfg.sh	Creates and populates the Dell utility partition and creates the deployment partition on a specified hard drive. Before doing so, however, this script cleans out all existing partitions in your system.
raccap.sh	Captures RAC settings to a file.
racrep.sh	Applies (replicates) the RAC settings saved in a file to the target system.
raidcfg.sh	Configures all RAID controllers detected in a system.
suseinst.sh	Performs an unattended installation of a supported SUSE Linux Enterprise Server operating system.
syscap.sh	Captures BIOS and BMC configuration settings and saves them to a file.
sysdrmk	Used by **upinit.sh** to patch the boot sector.
sysrep.sh	Applies (or replicates) BIOS and BMC configuration settings and configures the target system.
tkenvset.sh	Sets the path of the DTK installation and variables for the specific system being deployed.
upinit.sh	Creates and populates the Dell utility partition.

Documentation

Table 7-6 describes documents containing reference information for each DTK utility and instructions for using the DTK utilities and sample scripts. Documentation can be found at `mnt/cdrom/docs`.

Table 7-6. Documentation

Toolkit Components	Purpose
Command Line Interface Reference Guide	Describes all DTK utilities, including all command line arguments and values, syntax, help, and examples.
User's Guide	Contains best practices guidelines for performing a scripted deployment.
ReadMe	Provides the latest product information. It is located in the **mnt/cdrom/docs** directory.

Drivers and Dell Real Mode Kernel (DRMK)

Hardware application programing interface (HAPI) drivers are essential for DTK tools to work in the embedded Linux environment.

DRMK contains a file necessary to create a bootable Dell utility partition. This file is **mbr**..

Other Supporting Tools and System Files

While the DTK provides all the necessary utilities and scripts for deploying Dell systems, additional applications, utilities, and system files are needed to create the pre-operating system environment in which this toolkit is used. Table 7-7 is a matrix of system files and other tools that you need to use the DTK utilities and its sample scripts.

Table 7-7. Supporting Tools and System Files

Additional Tools	Purpose	Source	Save in DTK location
mkfs.*	To format hard-drive partitions.	Any system running a Linux operating system.	In the path.
fdisk	To partition the hard drive.	Any system running a Linux operating system.	In the path.
unzip	To unzip files in Linux.	Any system running a Linux operating system.	In the path.

Table 7-7. Supporting Tools and System Files *(continued)*

Additional Tools	Purpose	Source	Save in DTK location
upimg	Contains system-specific Dell utility partition files. **NOTE:** In the previous releases, each Dell system required a unique **upimg** file. Beginning this release, the **upimg** file is common for all Dell systems.	**/opt/dell/toolkit/systems,** the Dell Support website at **support.dell.com,** or the Dell Systems Build and Update Utility on the on the *Dell Systems Management Tools and Documentation* DVD.	**/opt/dell /toolkit/ systems**
kernel image	Used during a Linux unattended installation.	Available on the first disk of the Linux CD.	**/opt/dell /toolkit /systems/linux**
initial ramdisk	Used during a Linux unattended installation. **NOTE:** For legacy operating systems, you may need to load the driver disk. At the end of the Red Hat Enterprise Linux Server installation, you may need to install the drivers available on the Systems Build and Update Utility.	Available on the Linux operating system CD or on the Systems Build and Update Utility on the *Dell Systems Management Tools and Documentation* DVD.	**/opt/dell /toolkit /systems**

Setting Up a Directory Structure for Deployment

To perform a DTK-enabled deployment, you must create a deployment directory structure on either a network share or your local workstation hard drive. The deployment directory structure is a central repository for all deployment files and can be used successively as a testing space, a repository for a network deployment, and a repository to build a bootable deployment CD. If you plan to perform a network-based deployment, you must set up the directory structure on a network volume that is accessible from your workstation, the source system (to generate configuration information), and the target system (to deploy).

If you plan to run a scripted deployment from the DTK sample scripts, ensure that the following files are correctly structured in the deployment directory. The DTK sample scripts use this structure to set the default paths for deployment. If you make changes to the directory structure, you must also make changes to the sample scripts. The deployment directory structure consists of the following files:

- DTK utilities and files
- System-specific directories
- Operating system installation files directories

DTK Utilities and Files

All necessary DTK files are provided on the ISO image. The sample scripts provided with the DTK are set up to use the default paths.

NOTE: If you choose to create a directory structure for the DTK components that is different from the one provided in the initial installation, carefully review and edit each sample script (where necessary) to reflect the different directory structure.

NOTE: The unzip utility is needed to create the Dell utility partition. This utility should be copied into the **/systems** folder.

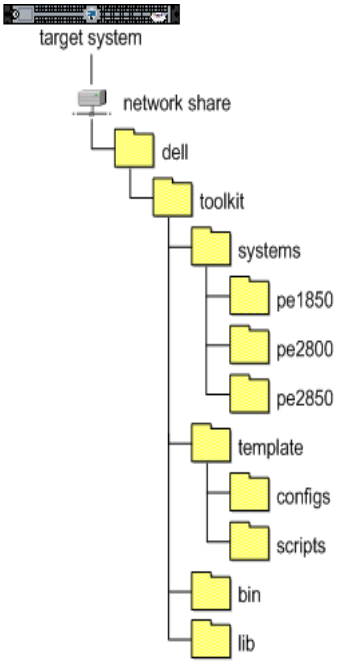
System-Specific Files

The directory **/opt/dell/toolkit/systems/system name** acts as a central repository for all the system-specific deployment information. The `/systems` folder acts a repository for the common files required for systems. For example, you can store all profile configuration files for an optimally configured PowerEdge™ 2850 in the directory **/opt/dell/toolkit/systems/pe2850**. See Figure 7-3. These files can then be accessed from this directory when deploying multiple PowerEdge 2850 systems. For more information about obtaining Dell utility partition files, see "How to Obtain Dell Utility Partition Files."

NOTE: You should mount the network share on to the target system in the master deployment file you created.

NOTE: If you choose to create a directory structure for the system-specific files that is different from the one suggested in this guide, be sure to carefully review and edit each sample script (where necessary) to reflect the different directory structure.

Figure 7-3. DTK System-Specific File Structure



Creating Operating System Installation Directories for Supported Linux Operating Systems

NOTE: The following procedures are provided as a best practices guideline for creating operating system distribution folders compatible with the DTK sample scripts. For complete instructions on setting up an installation directory structure for an unattended installation, see your operating system documentation.

For deployment of supported Linux operating systems, copy the ISO images for the operating system installation to a folder on your network (for example, **/linuxboot**). This folder should be accessible after DTK completes its tasks.

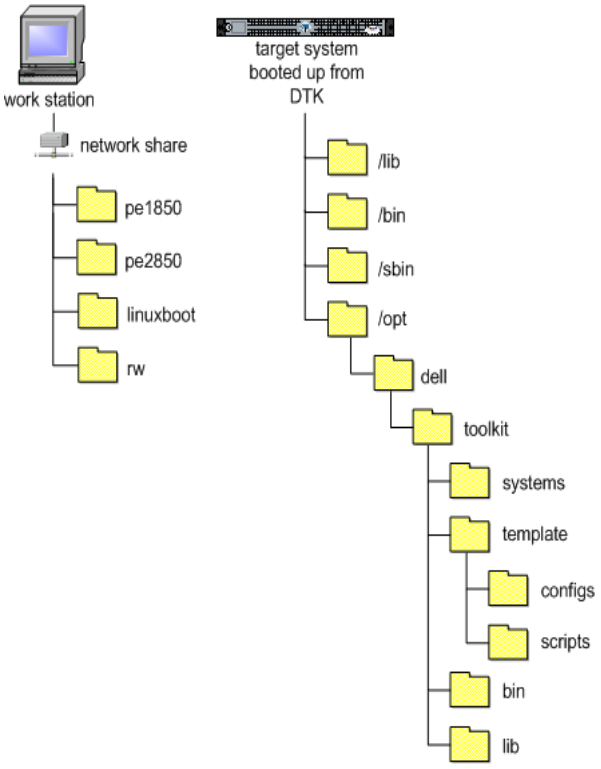
Also, copy the following into the **/linuxboot** folder:

- `kernel image` (from the Linux operating system CD)
- `initial ramdisk` from the Dell Systems Build and Update Utility on the *Dell Systems Management Tools and Documentation* DVD

See Figure 7-4 for a sample directory structure for deployment of a supported Linux operating system.

NOTE: If you choose to create a directory structure for the Linux installation files that is different from the one suggested in this guide, be sure to carefully review and edit each sample script (where necessary) to reflect the different directory structure.

Figure 7-4. Sample Operating System-Specific File Structure (Linux Operating System)



Configuring a Source System

You need an optimally configured system both for testing purposes and as the source for the SYSCFG and RAC configuration files to be used in the deployment process. See "Using the DTK Sample Scripts to Capture Configuration Information in Embedded Linux" for instructions on using the sample DTK scripts, **syscap.sh** and **raccap.sh**, to generate SYSCFG and RAC configuration files that are used to configure the target system.

You can use the Dell Systems Build and Update Utility and the Systems Service and Diagnostics Tools on the *Dell Systems Management Tools and Documentation* DVD supplied with your system to:

- Streamline the operating system installation
- Setup and configure your system, including automatic discovery and configuration of Dell-provided RAID controllers and network adapters
- Access and load the latest utilities, drivers, and diagnostics for your system
- Install Dell OpenManage™ systems management software

NOTE: You can also obtain the latest drivers from the Dell Support website at **support.dell.com.**

For complete information about using the Dell Systems Build and Update Utility and the Dell Systems Service and Diagnostics Tools, see the *Dell Systems Build and Update Utility Quick Reference Guide* and the *Dell Systems Service and Diagnostics Tools Quick Installation Guide* respectively, which are available on the Dell Support website at **support.dell.com** and on the *Dell Systems Management Tools and Documentation* DVD.

After you have loaded the operating system and all applicable device drivers from the Systems Build and Update Utility and Systems Service and Diagnostics Tools, check the Dell Support website at **support.dell.com** for the latest drivers. Put all necessary drivers to be used in the deployment in the system directory for the target system. For example, if your target system is a PowerEdge 1850, put all necessary files in the **/opt/dell/toolkit/systems/pe1850** directory.

How to Obtain Dell Utility Partition Files

Obtain the Dell utility partition files from **/opt/dell/toolkit/systems**, the Dell Support website at **support.dell.com**, or the Systems Build and Update Utility on the *Dell Systems Management Tools and Documentation* DVD.

RAC Configuration

Configure the RAC on your source system using the **racadm** utility, if applicable. See the *Dell Remote Access Controller Racadm User's Guide* for comprehensive information about using the **racadm** utility to configure the RAC on your source system.

After you have successfully prepared your directory structure for a deployment and fully configured your source system, you are ready to write, review, or edit the deployment sample scripts. For instructions on editing the sample scripts necessary to run a full deployment, see "Preparing the Script Files for Deployment Using Embedded Linux."

8

Preparing the Script Files for Deployment Using Embedded Linux

This section describes the steps required to use the Deployment Toolkit (DTK) sample scripts to perform a full deployment of a target system in the embedded Linux environment.

There are four main tasks:

- Writing the master deployment file and editing the **tkenvset.sh** script to set the variable values used in the deployment scripts
- Editing the supporting task scripts, as necessary
- Using the DTK sample scripts to capture basic input/output system (BIOS), baseboard management controller (BMC), and remote access controller (RAC) configuration information from an optimally configured source system
- Preparing the operating system installation scripts and answer files
- Configuring supported RAID controllers

Using the Sample DTK Scripts to Run a Full Deployment in Embedded Linux

The Master Deployment File

A scripted DTK deployment relies on the master deployment file to integrate all the applicable DTK task scripts and utilities and perform pre-operating system configuration tasks before launching the operating system installation files. The master deployment file, along with the **tkenvset.sh** scripts, make up the master set of instructions that drives the deployment process.

Editing the Sample Batch Files to Perform a Complete System Deployment in Embedded Linux

Using the Sample Scripts in a Scripted Deployment

The sample scripts utilize the DTK utilities, the additional supporting utilities, the source system configuration files, the target system hardware drivers, and the operating system installation files to perform a full deployment of the target system.

The sample scripts provided with the DTK can be edited and customized using any ASCII text editor that conforms to the hardware available on the target systems. While it is recommended that you use the DTK sample scripts to automate your deployment tasks, you can create your own system deployment batch files using any ASCII text editor by incorporating the various DTK utilities and Linux commands in the batch file.

NOTE: For more information about scripting batch files and specific batch file commands, see your Linux operating system documentation.

Each of the utilities, scripts, and system configuration files are described below. Depending on your requirements, you may use all or only some of them in your deployment:

The master deployment file— You can write your master deployment file to perform a full deployment of a system with the help of utilities, scripts, and configuration files. This file is not provided with the sample scripts.

You may need the following scripts in your deployment.

Table 8-1. Capture Scripts

Script Name	Description	Associated Files
raccap.sh	Captures RAC settings to a file for Dell™ Remote Access Controller (DRAC) 4.	• **tkenvset.sh** — Used by **raccap.sh** to inform the task scripts where to find the DTK utilities, scripts, and configuration files. • **racadm** — Used by **raccap.sh** to retrieve RAC information and to configure settings. The wrapper script, **racadm**, calls **racadm3** or **racadm4** depending on the system configuration. • **syscfg** — Used by **raccap.sh** to configure DRAC 5.
syscap.sh	Captures BIOS and BMC configuration settings and saves them to a file.	• **tkenvset.sh** — Used by **syscap.sh** to inform the task scripts where to find the DTK utilities, scripts, and configuration files. • **syscfg** — Used by **syscap.sh** to configure the BIOS and BMC settings.
raidcap.sh	Captures the RAID settings and saves them to a file.	• **tkenvset.sh** — Used by **raidcap.sh** to inform the task scripts where to find the DTK utilities, scripts, and configuration files. • **raidcfg**— Used by **raidcap.sh** to retrieve RAID information and to configure settings.

Table 8-2. Replication Scripts

Script Name	Description	Associated Files
racrep.sh	Applies the RAC configuration settings saved in a file to the current system.	• **tkenvset.sh** — Used by **racrep.sh** to inform the task scripts where to find the DTK utilities, scripts, and configuration files. • **racadm** — Used by **racrep.sh** to replicate RAC information and to configure settings. **racadm** is the wrapper script which calls **racadm3** or **racadm4** depending on the system configuration. • **syscfg** — Used by **racrep.sh** to configure DRAC5.
sysrep.sh	Applies the BIOS and BMC settings saved in a file to the current system.	• **tkenvset.sh** — Used by **sysrep.sh** to inform the task scripts where to find the DTK utilities, scripts, and configuration files. • **syscfg** — Used by **sysrep.sh** to configure the BIOS settings.

Table 8-3. Deployment Scripts

Script Name	Description	Associated Files
lininst.sh	Performs an unattended installation of a supported Red Hat® Enterprise Linux® Server operating system.	• **tkenvset.sh** — Used by **lininst.sh** to inform the task scripts where to find the DTK utilities, scripts, and configuration files. • **grub** — Used to install the operating system in conjunction with the **unattended answer** file. It also uses the following images in conjunction with the **unattended answer** file to perform unattended operations: – **initial ramdisk**— Available on the Dell Systems Build and Update Utility – **kernel image**— Available on the Linux operating system CD • **ks.cfg** — Used by **lininst.sh** in conjunction with **grub** to install the Red Hat Enterprise Linux Server (versions 3 and 4) operating system files. • **ks-rhel5.cfg** — Used by **lininst.sh** in conjunction with **grub** to install the Red Hat Enterprise Linux Server (version 5) operating system files.

Table 8-3. Deployment Scripts

Script Name	Description	Associated Files
suseinst.sh	Performs an unattended installation of a supported SUSE® Linux Enterprise Server operating system.	• **tkenvset.sh** — Used by **suseinst.sh** to inform the task scripts where to find the DTK utilities, scripts, and configuration files. • **grub** — Used to install the operating system in conjunction with the **unattended answer** file. It also uses the following images in conjunction with the **unattended answer** file to perform unattended operations: – **initial ramdisk**— Available on the SUSE Linux Enterprise Server operating system CD – **kernel image**— Available on the SUSE Linux Enterprise Server operating system CD • **autoinst.xml** — Used by **suseinst.sh** in conjunction with **grub** to install the operating system files.

Table 8-4. Other Scripts

Script Name	Description	Associated Files
tkenvset.sh	Sets the path of the DTK installation and variables for the system to be deployed and configured. The variables and paths specified in this file must be specified before any of the other supporting scripts can be used.	• **syscfg** — Used by **tkenvset.sh** to discover the system type.
raidcfg.sh	Configures the selected RAID controller detected in the system.	• **tkenvset.sh** — Used by **raidcfg.sh** to inform the task scripts where to find the DTK utilities, scripts, and configuration files. • **raidcfg** — Used by **raidcfg.sh** to retrieve RAID information and to configure settings.

Table 8-4. Other Scripts

Script Name	Description	Associated Files
partcfg.sh	Creates and populates the Dell utility partition and the deployment partition on a specified disk. The **partcfg.sh** script uses a `/tmp` folder to store a few temporary files critical to its execution.	• **tkenvset.sh** — Used by **partcfg.sh** to inform the task scripts where to find the DTK utilities, scripts, and configuration files. • **fdisk** — Used to create partitions on your disk. When creating an embedded Linux ISO image, ensure that **fdisk** is present in the embedded Linux you created. This script uses the following file for noninteractive (unattended) operations: – **upinit.sh**— Used by **partcfg.sh** to create and populate the Dell utility partition. – **upinit.sh**— Used to create and populate the Dell utility partition. **NOTE:** Do not make any modifications to the **upinit.sh** script.

Table 8-4. Other Scripts

Script Name	Description	Associated Files
upinit.sh	Creates and populates the Dell utility partition. Do not make any modifications to the **upinit.sh** script.	• **tkenvset.sh** — Used by **upinit.sh** to inform the task scripts where to find the DTK utilities, scripts, and configuration files. • **fdisk** — Used to create partitions on your disk. • **mbr** — File necessary to create a bootable Dell utility partition. • **sysdrmk** — Used by **upinit.sh** to patch the boot sector. • **unzip** — Used by **upinit.sh** to populate the Dell utility partition. • **mke2fs** — Used to make a **ext2** file system. • **upimg.bin** — Contains the zipped contents of the Dell utility partition.

Editing the tkenvset.sh Script

The majority of the editing required for the DTK sample scripts involves the following script. As a result, the majority of the scripting discussion focuses on **tkenvset.sh.** Other task scripts might require edits in certain situations. These optional edits are discussed in "Editing DTK Task Scripts."

tkenvset.sh

The **tkenvset.sh** script sets the paths for the DTK utilities and scripts and sets the variables for the systems to be deployed and configured. These variables and paths must be specified before any of the scripts can be used.

Specifying DTK Installation Paths

The **tkenvset.sh** sample script is provided with the following default values. These values must be confirmed, in some cases, and modified, in other cases, before a deployment. Locate the **Deployment Toolkit Installation Paths** heading to find the following lines:

```
DT_PATH=$dt_drive/opt/dell/toolkit

DT_TOOLS=$dt_path/bin

DT_SYSTEMS=$dt_path/systems

DT_SCRIPTS=$dt_path/template/scripts

DT_STOP_ON_ERROR=FALSE

DT_DRMK=$dt_path/opt/dell/drmk
```

Perform the following steps to confirm and/or edit the installation path variables:

1. Set the variable that defines the path to the full set of DTK deployment files (`DT_PATH`). This variable is set to `$dt_drive/opt/dell/toolkit` by default in the sample script. Edit this path to reflect the actual location of the DTK deployment files, if necessary.
2. Set the variable that defines the directory containing the DTK utilities (`DT_TOOLS`). This variable is set to `$dt_path/bin` by default in the sample script. Edit this path to reflect the actual location of the DTK utilities, if necessary.
3. Set the variable that defines the directory containing the configuration files for your Dell systems (`DT_SYSTEMS`). This variable is set to `$dt_path/systems` by default in the sample script. Edit this path to add a subdirectory for the Dell system you plan to deploy. For example, `$dt_path/systems/pe1850`.
4. Set the variable that defines the directory containing the DTK deployment scripts (`DT_SCRIPTS`). This variable is set to `%DT_PATH%\Template\Scripts` by default in the sample script. Edit this path to reflect the actual location of the DTK deployment scripts, if necessary.

5 Set the variable that indicates whether you want the deployment process to exit when any error is returned (`DT_STOP_ON_ERROR`). This variable is set to `FALSE` by default in the sample script, indicating that errors will be handed by the **errhndl.sh** scripts. Set the value to `TRUE` only if you want the deployment to exit on any error.

6 Set the variable that defines the directory containing the DRMK system files necessary to create a Dell utility partition (`DT_DRMK`) . This variable is set to `$dt_path/opt/dell/drmk` by default in the sample script. Edit this path to reflect the actual location of the DRMK files, if necessary.

Specifying the SYSCFG Variables

The SYSCFG replication file is set to **syscfg.ini** by default. If you used the **syscap.sh** sample script to generate the **syscfg.ini** file, you do not have to edit these default settings. If you have made any modifications to the sample scripts or to the suggested directory structure, you must confirm that the **syscfg.ini** file is correctly named and that the path is set correctly for your deployment.

To edit the SYSCFG capture variables, locate the following lines in **syscap.sh**:

```
set DT_SYS_CAPFILE=
$dt_systems/$dt_platform/syscfg.ini
```

To edit the SYSCFG replication variables, locate the following lines in **sysrep.sh**:

```
DT_SYS_REPFILE=$dt_systems/$dt_platform/syscfg.ini
```

NOTE: If the replication requires a change in the memory redundancy mode, the target server should be rebooted and the script should be executed again to complete the replication.

Specifying the RAC Configuration Variables

The IP address for your target system RAC is set in the sample script to `10.98.8.121` by default. Edit this value to reflect the actual IP address of your target system RAC, if applicable. Additionally, the RAC configuration file used to configure your RAC is set by default to **raccfg.ini**. If you use the **raccap.sh** sample script to generate the **raccfg.ini** file, you do not have to edit this default setting. If you have made any modifications to the sample scripts or to the suggested directory structure, you must confirm that **raccfg.ini** is correctly named and that the path is correctly set for your deployment.

To edit the RAC configuration variables, locate the following lines in **raccap.sh**:

```
DT_RAC_CAPFILE=$dt_systems/$dt_platform/raccfg.ini
```

To edit the RAC configuration variables, locate the following lines in **racrep.sh**:

```
# RAC section

DT_NICCFG=10.98.8.121

set DT_RAC_CAPFILE=
$dt_systems/$dt_platform/raccfg.ini
```

Specifying Variables Used to Create Disk Partitions and Install the Dell Utility Partition

NOTE: If the primary deployment state (`DT_PARTN`) is set to `off`, the partitioning deployment state subsection is skipped.

The **partcfg.sh** sample script provides the following disk partitioning default values. These values must be confirmed, in some cases, and modified, in other cases, before performing a deployment. Locate the **partn** label to find the following lines in **partcfg.sh**:

```
DT_PARTN_UP=on

DT_MOUNT=/tmp/up


# Default Hard Drive

DT_HD=/dev/sda


# PARTCFG section

set DT_OS_SIZE=4000

# DELLUP section

set DT_UP_SIZE=32

set DT_UP_IMAGE=$dt_systems/upimg.bin
```

```
# RAIDCFG section
DT_VD_SIZE=10480
# default size for virtual disk
```

CAUTION: `DT_MOUNT` **specifies the mount point of the Dell utility partition. If your RAMDRIVE, hard drive, or other device already owns this designation, mount will not override the setting and the utility partition will not be created.**

Perform the following steps to edit the default partition configuration default values, as necessary:

1. Edit the default utility partitioning variable (`DT_PARTN_UP=on`), if necessary. This variable sets whether you want to install the Dell utility partition.

   NOTE: Dell recommends that you install the Dell utility partition.

2. Edit the default variable value for the default hard drive (`DT_HD=sda`), if applicable.
3. Edit the default variable value (in MB) for the primary operating system partition (`DT_OS_SIZE=4000`), if applicable.
4. Edit the default variable value for file system type (`DT_OS_FSTYPE=FAT32`), if applicable.
5. If the `DT_PART_UP` variable is set to `on`, edit the default variable value (in MB) for the utility partition (`DT_UP_SIZE=32`), if applicable.
6. If the `DT_PART_UP` variable is set to `on`, edit the default variable value for the utility partition installation package location (`DT_UP_IMAGE=$dt_systems/upimg.bin`), if applicable.

   You should have already copied the **upimg.bin** file to the proper directory, as instructed in "Preparing the Directory Structure for Deployment in Embedded Linux." Edit this variable only if you have changed the name or path of the utility partition installation package.

Editing DTK Task Scripts

The task scripts called by the master batch file do not require any edits (with the following exceptions) if the default variable values are utilized in the **tkenvset.sh** scripts. However, if you make any changes to the default values set by the **tkenvset.sh**, you must make the same edits to the matching default values set in the appropriate task scripts. Locate the "USER MODIFICATION REQUIRED" heading in the task scripts to find the variable values you might need to edit. Each of the following task scripts can be called by the master batch script during a full deployment:

- **racrep.sh**
- **raidcfg.sh**
- **sysrep.sh**
- **partcfg.sh**

racrep.sh

The **racrep.sh** sample script applies RAC configuration options to the target system based on configuration information that has been saved in a configuration file. User input is optional, depending on variables set in the **racrep.sh** sample script. This script uses the **racadm** wrapper script or the **syscfg** utility (depending on the version of RAC on your system) to configure the RAC in the target system using the RAC configuration file, **raccfg.ini**, passed as the first parameter to this script. If this parameter is not passed, the default variables values used in this script are set in the **tkenvset.sh** scripts.
An optional second parameter can be passed to this file, which specifies the IP address to be used to configure the RAC. The **raccfg.ini** file is generated with the **raccap.sh** sample script. See "raccap.sh" for information about using the **raccap.sh** sample script to capture RAC configuration information.

NOTE: For *x9xx* or later systems, **racrep.sh** and **sysrep.sh** have duplicate functions. If you run **racrep.sh** followed by **sysrep.sh**, the latter overwrites the former's settings.

raidcfg.sh

The **raidcfg.sh** sample script configures RAID controllers detected on the system. User input is optional to run this script. The parameter is a name or path to a log file. This parameter is not set to any value by default, and no log file is generated. The default variable values used in this script are set in the **tkenvset.sh** scripts. This script uses the **raidcfg.sh** utility to automatically configure the detected controllers on your target system. For example, for the first controller discovered, if **raidcfg.sh** discovers only one attached hard drive, the script tries to create RAID 0; if two hard drives are discovered, a RAID 1 configuration is enabled; if three or more hard drives are discovered, a RAID 5 configuration is enabled.

NOTE: On the detected controller, a RAID virtual disk is created only if the array disks connected to that particular controller have space available and, where applicable, are not part of an already existing array group.

sysrep.sh

The **sysrep.sh** sample script applies SYSCFG options to the target system based on configuration information that has been saved in a configuration file. User input is optional to run this script, depending on the variable settings in the **sysrep.sh scripts**. This script uses the **syscfg** utility to configure the BIOS and BMC in the target system using the configuration file, **syscfg.ini**, passed as the first parameter to this script. If this parameter is not passed, the default variables values used in this script are set in the **sysrep.sh** scripts. The **syscfg.ini** file is generated with the **syscap.sh** sample script. See "syscap.sh" for information about using the **syscap.sh** sample script to capture BIOS and BMC settings. The second parameter is a name or path to a log file. This parameter is not set to any value by default, and no log file is generated.

NOTE: If the replication requires a change in the memory redundancy mode, the target server should be rebooted and the script should be executed again to complete the replication.

NOTE: For *x9xx* or later systems, **racrep.sh** and **sysrep.sh** have duplicate functions. If you run **racrep.sh** followed by **sysrep.sh**, the latter overwrites the former's settings.

partcfg.sh

The **partcfg.sh** sample script creates and populates the Dell utility partition using **upinit.sh** and creates the deployment partition on a specified disk. (See "upinit.sh" for more information about using the **upinit.sh** tool.) User input is not required to run this script. The default variable values used in this script are set in the **tkenvset.sh** scripts.

NOTE: Ensure that DT_HD is set to the required disk to be configured before you execute the **partcfg.sh** script.

Using the DTK Sample Scripts to Capture Configuration Information in Embedded Linux

The **syscfg** and **racadm** utilities can read the source system configuration and duplicate that configuration on a target system using a sample DTK script file. These configuration files are used during a full deployment to configure the target system's BIOS, BMC, and RAC settings. The following sections provide instructions for using the **syscap.sh** and **raccap.sh** sample scripts to create configuration files.

Capturing the System BIOS and BMC Configuration With the syscap.sh Sample Script

You can replicate the system BIOS and BMC configurations from a source Dell system to an identical target Dell system. For example, from one PowerEdge™ 1850 to another PowerEdge 1850).

NOTE: For the script to complete successfully, the source system must have access to the network share that includes the prebuilt DTK directory structure.

syscap.sh

The **syscap.sh** sample script captures BIOS and BMC configuration settings and saves them to a specified location. An optional user input can be passed as a parameter. You can provide the path or file name as an input to this script. The default variable values used in this script are set in the script to the correct default values to perform a scripted deployment. The **syscap.sh** script uses the **syscfg** utility to retrieve BIOS and BMC configuration settings from the source system and to save the settings in a file named **syscfg.ini** in the `Z:\Dell\Toolkit\Systems\<target system>` directory.

Perform the following steps to run the **syscap.sh** sample script to capture BIOS and BMC configuration information in the **syscfg.ini** file:

1. Edit the start-up script file on your embedded Linux image to call the **syscap.sh** script after the network share with the prebuilt DTK directory structure is mounted. For example, add:

   ```
   source/opt/dell/toolkit/template/scripts
   /syscap.sh
   ```

2. Boot the source system into embedded Linux.

   The script runs and saves configuration information in the **/opt/dell/toolkit/systems/<target system>** directory.

3. Edit the start-up script file in your embedded Linux image to remove the text you added in step 1.

NOTE: For *x9xx* or later systems, **raccap.sh** and **syscap.sh** have duplicate functions. If you run **raccap.sh** followed by **syscap.sh**, the latter overwrites the former's settings.

Capturing the RAC Configuration With the raccap.sh Sample Script

If your system has a RAC, you can replicate the RAC configuration from a source Dell system's RAC to an identical target Dell system with an identical RAC. For example, from one PowerEdge 1850 with a DRAC4/I to another PowerEdge 1850 with a DRAC4/I.

NOTE: For the script to complete successfully, the source system must have access to the network share that includes the prebuilt DTK directory structure.

raccap.sh

The **raccap.sh** sample script captures RAC configuration settings and saves them to a specified location. An optional user input can be passed as a parameter. You can provide the path or filename as an input to this script. The default variable values used in this script are set in the script to the correct default values to perform a scripted deployment. The **raccap.sh** script uses the **racadm** wrapper script or **syscfg** utility (depending on the RAC version on your system) to retrieve RAC configuration settings from the source system and to save the settings in a file named **raccfg.ini** in the **/opt/dell/toolkit/systems/<target system>** directory.

Perform the following steps to run the **raccap.sh** sample script to capture RAC configuration information in the **raccfg.ini** file:

1. Edit the start-up script file on your embedded Linux image to call the **raccap.sh** script after the network share with the prebuilt DTK directory structure is mounted. For example, add:

   ```
   source/dell/toolkit/template/scripts/raccap.sh
   ```

2. Boot the source system into embedded Linux.

   The script runs and saves configuration information in the **/opt/dell/toolkit/systems/<target system>** directory.

3. Edit the start-up script file on your embedded Linux image to remove the text you added in step 1.

NOTE: The RAC configuration file is referred to as the **.cfg** file in the *Dell Remote Access Controller Racadm User's Guide and the Dell Remote Access Controller 4 User's Guide*. See these guides for additional information about manually creating a RAC configuration file.

NOTE: Do not replicate the RAC IP address when creating a **.cfg** file. Replicating the RAC IP address can leave the system inaccessible because multiple systems are configured with the same IP address.

NOTE: For *x9xx* or later systems, **raccap.sh** and **syscap.sh** have duplicate functions. If you run **raccap.sh** followed by **syscap.sh**, the latter overwrites the former's settings.

upinit.sh

The **upinit.sh** tool is used to create and populate the Dell utility partition.

NOTE: You can use this script to create a new Dell utility partition only if there are no existing partitions on your hard disk.

You can also use this script to upgrade or downgrade an existing Dell utility partition. Upgrades or downgrades can be made only to existing Dell utility partitions to capture BIOS and BMC settings.

CAUTION: The existing Dell utility partition should be the first primary partition on the hard disk. Also, the Dell utility partition must be greater than or equal to 32 MB and less than 2 GB.

For details on the command line arguments to be used for **upinit.sh**, see the *Deployment Toolkit Command Line Interface Reference Guide*.

Preparing the Operating System Installation Scripts

Red Hat Enterprise Linux Server Operating Systems

The **lininst.sh** script is used to install supported Red Hat Enterprise Linux Server operating systems.

NOTE: While installing Linux, ensure that you install **grub** in the boot partition. Otherwise, you will not be able to boot to the utility partition by pressing the **F10** key during reboot.

lininst.sh

The **lininst.sh** sample script is used to perform an unattended installation of a supported Red Hat Enterprise Linux Server operating system. The variables and paths must be specified before the operating system can be installed.

CAUTION: It is strongly recommended that you consult your Red Hat Enterprise Linux Server operating system documentation and unattended deployment documentation to develop a thorough understanding of the unattended installation process before attempting to perform a full scripted deployment.

Utilities Used

grub — Used in conjunction with the **kernel image** and **initial ramdisk** files, along with the **unattended** answer file to install the operating system.

Variables Used

- `DT_OS_DISK` — Sets the disk on which to deploy the Red Hat Enterprise Linux Server operating system.
- `DT_OS_SRC` — Sets the location from where to copy the **kernel image** and **initial ramdisk** files.

External Dependencies

The `DT_PLATFORM` variable is set during deployment by **tkenvset.sh** using platform discovery.

The **lininst.sh** script uses the **unattended answer** file in conjunction with the **grub** utility and **kernel image** and **initial ramdisk** files to install the operating system files. See "ks.cfg and ks-rhel5.cfg" for more information about preparing the **unattended answer** file.

ks.cfg and ks-rhel5.cfg

Depending on the version of the Red Hat Enterprise Linux Server operating system, the **ks.cfg** or **ks-rhel5.cfg** files are the default names of the answer files that you use to automate an unattended Red Hat Enterprise Linux Server installation. A sample copy of the **ks.cfg/ ks-rhel5.cfg** files are provided as part of the DTK download in the **/opt/dell/toolkit/template/configs** directory. The **ks.cfg/ks-rhel5.cfg** files contain the headings and parameters that instruct the installation utility to perform various configuration tasks. Before a full deployment can be run, you must edit all appropriate headings and parameters, as necessary.

NOTE: The sample file provided to install Red Hat Enterprise Linux Server (version 5) is named **ks-rhel5.cfg**. To use the sample **ks-rhel5.cfg** file to install Red Hat Enterprise Linux Server (version 5) using **lininst.sh**, ensure that you rename the file to **ks.cfg**.

NOTE: See your operating system documentation for instructions on modifying the options in the **ks.cfg/ks-rhel5.cfg** file to customize the scripted installation of your operating system.

When you have completed all necessary modifications of the **ks.cfg/ ks-rhel5.cfg** file, save the completed file in the **/opt/dell/toolkit/systems/<target system>** directory.

Install Linux Using lininst.sh

On a Linux system, edit the **/etc/exports** file. If the folder you want to share is **/home**, then **/etc/exports** file should contain the "**/home *(rw)**" line. Restart the NFS Service on your system and copy the relevant OS images to the NFS share.

Perform the following steps to install Red Hat Enterprise Linux (32-bit) using the **lininst.sh** script:

1. Create the virtual disk using **raidcfg** command.
2. Run the **partcfg.sh** script before running **lininst.sh** to create Dell Utility partition and OS partition.
3. Create a directory under **/tmp**. for example **os_src**.
4. Mount your NFS Share to **/tmp/os_src**.
5. Copy the file **initrd.img** for RHEL 32-bit to the NFS Share.
6. Copy **vmlinuz** from RHEL 32-bit 1st CD to the NFS Share.
7. Copy the **ks.cfg** file from **/opt/dell/toolkit/template/configs** to the NFS Share.

8 Edit the file **ks.cfg**. Set the NFS IP Address where RHEL 32-bit images are available and the NFS share path.

9 Run **export DT_OS_DISK=/dev/sda**, the disk device on which the Dell Utility partition was created.

10 Run **export DT_OS_SRC=/tmp/os_src**, where **vmlinuz**,**initrd.img** and **ks.cfg** available.

11 Run export **DT_HD=/dev/sda**.

12 From **/opt/dell/toolkit/template/scripts**, run **./lininst.sh** script.

13 After 20 seconds, the server reboots and boots to the GRUB prompt. The the RHEL 32-bit operating system unattended installation proceeds.

SUSE Linux Enterprise Server Operating Systems

The **suseinst.sh** script is used to install supported SUSE Linux Enterprise Server operating systems.

NOTE: While installing Linux, ensure that you install **grub** in the boot partition. Otherwise, you will not be able to boot to the utility partition by pressing the **F10** key during reboot.

suseinst.sh

The **suseinst.sh** sample script is used to perform an unattended installation of a supported SUSE Linux Enterprise Server operating system. The variables and paths must be specified before the operating system can be installed.

CAUTION: It is strongly recommended that you consult your SUSE Linux Enterprise Server operating system documentation and unattended deployment documentation to develop a thorough understanding of the unattended installation process before attempting to perform a full scripted deployment.

Utilities Used

grub — Used in conjunction with the **kernel image** and **initial ramdisk** files, along with the **unattended** answer file to install the operating system.

Variables Used

- `DT_OS_DISK`— Sets the disk on which to deploy the SUSE Linux Enterprise Server operating system.
- `DT_OS_SRC`— Sets the location from where to copy the **kernel image** and **initial ramdisk** files.
- `DT_OS_PART`— Sets the valid partition to install the **grub** files.
- `DT_OS_IMG_PATH`— Sets the network share where the operating system files are located.
- `DT_SUSE_AUTOINST`— Sets the network share from where the **autoinst.xml** file can be picked up.

External Dependencies

The `DT_PLATFORM` variable is set during deployment by **tkenvset.sh** using platform discovery.

The **suseinst.sh** script uses the **unattended answer** file in conjunction with the **grub** utility and **kernel image** and **initial ramdisk** files to install the operating system files. See "autoinst.xml for SUSE Linux Enterprise Server" for more information about preparing the **unattended answer** file.

autoinst.xml for SUSE Linux Enterprise Server

The **autoinst.xml** file is the default name of the answer file that you use to automate an unattended SUSE Linux Enterprise Server installation. A sample copy of the **autoinst.xml** file is provided as part of the DTK download in the **/opt/dell/toolkit/template/configs** directory. The **autoinst.xml** file contains the headings and parameters that instruct the installation utility to perform various configuration tasks. Before a full deployment can be run, you must edit all appropriate headings and parameters, as necessary.

NOTE: See your operating system documentation for instructions on modifying the options in the **autoinst.xml** file to customize the scripted installation of your operating system.

When you have completed all necessary modifications of the **autoinst.xml** file, save the completed file in the **/opt/dell/toolkit/systems/<target system>** directory.

After you have successfully edited your deployment scripts, you are ready to create your deployment media and run your deployment. For instructions on creating deployment media and running a full deployment, see "Running Deployment Scripts Using DTK and Embedded Linux."

9

Running Deployment Scripts Using DTK and Embedded Linux

This section provides best practices procedures and scenarios for using the Deployment Toolkit (DTK) to perform pre-operating system configuration tasks and to install supported operating systems on supported Dell™ systems.

After you have populated the deployment directory structure with all necessary files and carefully edited the deployment scripts and configurations files, you are ready to begin the final stage of the deployment process. Before running the deployment, however, you need to create specific deployment media to facilitate your chosen deployment method. The common deployment scenarios are:

- Deployment using Dell-provided embedded Linux
- Deployment using customized embedded Linux
- Using a third-party deployment solution framework

Deployment Using Dell-Provided Embedded Linux

There are two common scenarios:

- Network-based deployment
- CD-based deployment

Network-Based Deployment

The procedure below enables you to boot DTK over network and pass parameters (startup script location and name) during boot using the Pre-boot eXecution Environment (PXE) configuration file. This will automatically launch the start-up script off the network share, when DTK is booted.

1. Obtain the ISO image of the embedded Linux available on the Dell Support website at **support.dell.com**.
2. Burn the ISO image using any commonly available CD burning software.

   NOTE: When you create a bootable CD, this image is supplied to the CD burning software along with the complete deployment directory structure, which includes all necessary supporting files to perform the deployment.

   The CD burning software creates a self-bootable CD when the CD is booted.

3 Set up the Trivial File Transfer Protocol (TFTP) services and the Dynamic Host Configuration Protocol (DHCP) services on your network to boot the target system using Pre-Boot Execution Environment (PXE).

4 Copy the contents of the embedded Linux DTK ISO image (**SA.1**, **SA.2**, and **isolinux.cfg**) from **/mnt/cdrom/isolinux** to the `tftp` folder.

5 The **isolinux.cfg** file has various boot options. Select the method you want, and edit the **isolinux.cfg** file for the following:

- IP address of the network share
- Share path of the network share
- Name of your startup script

6 The **isolinux.cfg** file is ready for PXE-booting. Copy this **isolinux.cfg** file into the PXE configuration folder.

7 Create your own start-up script and place it in the network share specified in your modified **isolinux.cfg** file.

8 The modified script is picked up and executed during the boot process.

NOTE: The network share on which you have the scripts and DTK files are mounted to **/opt/dell/toolkit/systems**.

NOTE: Ensure that you specify the correct RAMDISK size in the **pxe config** file before booting through PXE. (A sample **pxe config** file is provided in **/mnt/cdrom/isolinux/isolinux.cfg**.)

CD-Based Deployment

1 Obtain the ISO image of the embedded Linux available on the Dell Support website at **support.dell.com**.

2 Burn the ISO image using any commonly available CD burning software.

NOTE: When you create a bootable CD, this image is supplied to the CD burning software along with the complete deployment directory structure, which includes all necessary supporting files to perform the deployment.

The CD burning software creates an image of the self-bootable ISO image and the deployment files on the CD.

3 Extract the contents of the ISO image to a folder on your hard drive.

4 Copy your custom scripts into the folder you created in step 3. (See "Installation for Embedded Linux" for an example of the folder structure.)

NOTE: Your scripts should also take care of copying miscellaneous items to unlock the CD, so that the CD can be mounted and ejected as required by the operating system's installation process.

5 In **/mnt/cdrom/isolinux.cfg**, edit the *cd install* section to point to your customized start-up script. In other words, add the `share_script` option; for example:

```
label 1
kernel SA.1
append initrd=SA.2 ramdisk_size=55321
Stage3_type=cdrom DEBUG=0 quiet share_type=cdrom
share_script=dracupdate.sh
```

Then, use `mkisofs` to make your customized CD image; for example:

```
mkisofs -o usaf.iso -b isolinux/isolinux.bin -c
isolinux/boot.cat -no-emul-boot -boot-load-size 4
-iso-level 2 -relaxed-filenames -r -J -pad -boot-
info-table /root/usaf/
```

NOTE: It is important to use `mkisofs` to make your customized image because the ISO Linux recognizes only the **isolinux.cfg** (and all files within **/isolinux**). If you do not use `mkisofs`, DTK's **openipmi** and HAPI libraries will not load and most of the Dell Update Packages will not work.

NOTE: The scripts that you copy into the CD will be copied to the RAM disk and run from the RAM disk. This task is done to ensure the CD is not locked. Ensure that your sample scripts have valid path names.

6 Copy the directory structure created on your work station into the root folder that you created in step 3. This folder contains the DTK CD, files necessary for operating system installation, and files required for replication.

7 Burn the contents of the folder you created in step 3 to a CD. (Use the **isolinux** utility to make the CD bootable.)

8 Your ISO image is ready for booting.

Deployment Using Customized Embedded Linux

Ensure that you have the following basic libraries, Dell toolkit libraries, tools, and utilities required for DTK to work in your customized Linux deployment environment (see Table 7-1 for details on files and RPMs to customize your embedded Linux):

- Drivers for all your hardware installed in your embedded Linux (from the Dell Support website at **support.dell.com**)
- Serial port configuration utilities (**setserial** and **stty**) for **racadm**
- Installed and working instrumentation drivers (from **/mnt/cdrom/tools**)
- Point-to-Point Protocol (PPP) tools
- Open Intelligent Platform Management Interface (IPMI) drivers
- Basic libraries and utilities for Linux to execute customized scripts
- DTK tools and utilities extracted from the embedded Linux CD (**/mnt/cdrom/tools/dell.tar.gz)**
- Necessary entries in the **ld.so.config** file so that the libraries get loaded

Integrate all the above mentioned libraries, Dell toolkit libraries, tools, and utilities into your embedded Linux environment and proceed with deployment.

NOTE: Ensure that the **raidcfg** executable in the **/bin** directory is a symbolic link to the **raidcfg** in the Dell-provided **/lib** folder.

NOTE: Refer to **start-stage3.sh**, **start-hapi.sh**, or **start-raid.sh** on **/mnt/cdrom/tools** to see how Dell-provided utilities and drivers are loaded.

NOTE: The **/opt/dell/srvadmin/shared** and **/opt/dell/srvadmin/hapi** directories should have Read-Write permissions. Refer to **start-stage3.sh** and **start-hapi.sh** for details.

Using a Third-Party Deployment Solution Framework

You can use the DTK with any existing third-party deployment solution framework that provides a PXE booting infrastructure that can be used as the transport mechanism for the DTK utilities. Because each third-party deployment framework is unique, however, these solutions fall outside the scope of this document. If you plan to utilize a third-party deployment solution framework, keep in mind that the DTK is an embedded Linux-based set of tools and scripts, so the deployment solution framework must also support embedded Linux as a pre-operating system environment.

10

Running Dell Update Packages in Embedded Linux

This section provides best practice procedures and scenarios for running Dell™ Update Packages in the embedded Linux environment on supported Dell systems.

The common scenarios for running Update Packages are:

Running Update Packages in Dell-Provided Embedded Linux

1. Obtain the required update packages from the *Dell Server Updates* DVD or from the Dell Support website at **support.dell.com**.
2. Save the update packages on a network share.
3. Mount the network share where you saved the update packages and run the individual packages.

NOTE: See the *Dell Update Packages for Linux Operating Systems User's Guide* for help on using Dell Update Packages and information on error codes.

Running Update Packages in Customized Embedded Linux

Before running update packages in your customized embedded Linux environment, ensure that you meet the dependencies listed in Table 10-1.

Table 10-1. Dependencies for Dell Update Packages to Run in Customized Embedded Linux

Update Packages	Dependency	Dependencies Available On
BIOS	Instrumentation drivers	**/mnt/cdrom/tools**
	dell_rbu	Kernel
ESM	OpenIPMI	**/mnt/cdrom/tools**
	Instrumentation drivers	**/mnt/cdrom/tools**

Table 10-1. Dependencies for Dell Update Packages to Run in Customized Embedded Linux *(continued)*

Update Packages	Dependency	Dependencies Available On
PERC	/etc/dataeng	/mnt/cdrom/tools
	megaraid driver 2.4.2.0	/mnt/cdrom/tools
	SCSI drivers	Kernel
RAC3	/etc/racser	/mnt/cdrom/tools
	PPP	/mnt/cdrom/tools
	racser3.sh	/mnt/cdrom/tools
RAC4	Open IPMI	/mnt/cdrom/tools
	Instrumentation drivers	/mnt/cdrom/tools
	racser4.sh	/mnt/cdrom/tools
RAC 5	Open IPMI	/mnt/cdrom/tools
	Instrumentation drivers	/mnt/cdrom/tools
iDRAC	Open IPMI	/mnt/cdrom/tools
	Instrumentation drivers	/mnt/cdrom/tools
Dell PowerVault™ 220S	SCSI drivers	Kernel
SAS	SAS drivers	Kernel
Any Update Package	RPM	Linux CD or any open source website
	sysvinit	
	grep	
	sed	
	awk	
	less	
	fmt	
	tar.gz	

After ensuring that you have met all the dependencies, run the update packages in your customized embedded Linux environment as follows:

1. Obtain the required update packages from the *Dell Server Updates* DVD or from the Dell Support website at **support.dell.com**.
2. Store the update packages on a network share.
3. Mount the network share where you stored the update packages and run the individual packages.

NOTE: For help on usage of Dell Update Packages and information on error codes, see the *Dell Update Packages for Linux Operating Systems User's Guide*.

11

Known Issues and Frequently Asked Questions

This section describes known issues with the Deployment Toolkit (DTK) utilities and scripts, including answers to some frequently asked questions.

Known Issues

The following issues are organized by the DTK utility or other function.

General Issues

- Virtual disk creation and/or deletion takes a long time on Microsoft® Windows® PE.

RAIDCFG Issues

- If you have created Redundant Array of Independent Disks (RAID) 1 with background initialization operation on CERC SATA 1.5/2S or CERC SATA 1.5/6CH you cannot delete the virtual disk using RAIDCFG, until the background initialization operation is completed. You must go to the controller basic input/output system (BIOS) to delete the virtual disk and then create another virtual disk using another option.
- If you delete a virtual disk on PowerEdge™ Expandable RAID controller (PERC) 4/IM on Windows PE, you need to reboot the system.
- Due to a rounding limitation, for RAID 0, RAID 1, and RAID 5, RAIDCFG can accept a virtual disk size that is 1 MB greater than the maximum allowable virtual disk size when creating a virtual disk. However, RAIDCFG will create the maximum virtual disk size and not give an error. If you need the maximum allowable virtual disk size, it is recommended that you do not provide the size in the CLI and let RAIDCFG calculate the size for the RAID type.
- Before creating a virtual disk using RAIDCFG on Cost-Effective RAID Controller (CERC) Serial Advanced Technology Attachment (SATA) 1.5/2S and CERC SATA 1.5/6CH, initialize the array disk(s) through the controller BIOS.

SYSCFG Issues

- Setup passwords and system passwords cannot be cleared using DTK.
- The Demand-based Power Management (DBPM) option does not work on PowerEdge 1855 systems with BIOS versions earlier than A05.
- On PowerEdge 1950, 2950, and PowerVault NX 1950 systems with internal USB, you can configure BIOS settings for internal USB Port only if the **User Accessible/external** (UA) USB ports are set to **All Ports On**. If the UA USB ports are not set to **All Ports On** and you try to configure BIOS setting for the internal USB port using SYSCFG, the configuration will appear to be successful but the changes will not take effect after the next reboot. If you set the UA USB port from **All Ports On** to **All Ports Off** or **Only Back Ports On**, the USB port will be automatically Set to **Off** during the next reboot.

RACADM Issues

- Dell™ Remote Access Controller (DRAC) III, DRAC III/XT, embedded remote access (ERA), ERA/O, and ERA/MC controllers are not supported on Windows PE.

Windows PE Installation Issues

- If virtual flash is enabled and does not contain a valid image (for example, if the virtual flash contains a corrupt or random image), you may not be able to install Windows 2000 or Windows Server® 2003 locally or remotely. To fix this issue, install a valid image on virtual flash or disable virtual flash if it will not be used during the installation procedure.

Frequently Asked Questions

The following questions are organized by the DTK utility or other function.

General Deployment Questions

Q: While running **SYSCFG.EXE** on a Windows PE image, the **SYSCFG.EXE** tool silently fails.

A: Ensure that you have built your Windows PE image with the `/WMI` option. See "Running Deployment Scripts Using DTK and Microsoft Windows PE."

Q: When I use the sample DTK scripts to deploy multiple systems, do I need to edit the configuration files to reflect unique information (such as unique system names, IP address, and BIOS asset tags) for each system?

A: The sample scripts are provided as examples for users who want to develop their own deployment process. You may find that the scripts work perfectly in your environment, or you may need to develop your own scripts entirely from scratch. If you are deploying multiple systems, for example, you will need to provide unique information for each system when appropriate. To perform this task, you will need to modify each **.ini** (or other configuration script) file to reflect the unique information for each system you are deploying (such as the remote access controller (RAC) IP addresses and BIOS asset tags). There are many options available to optimize this process and it is suggested that you conduct an Internet search for available tools.

Q: When I change my hard drive controller from SCSI to RAID (or RAID to SCSI), the system prompts me for confirmation during POST. How can I stop this from happening?

A: Use the **--noraidprompt** option with the **--embscsiraid** option to prevent the system from prompting during POST. Data loss will result from changing the state of the disk controller, so you should be certain before skipping the prompt. There is no method for using the **--noraidprompt** in the input file. If you want to change the controller state through an input file, you must configure your script to call the **SYSCFG.EXE** utility twice, once to specify the input file, and once to change the controller state with the **--noraidprompt** option.

Q: What do I do if the PARTCFG script fails?

A: Ensure that `DT_HD` is set to the required disk to be configured before you execute the PARTCFG script.

RAIDCFG Questions

Q: Why does RAIDCFG not give an error message when I use invalid **read**, **write**, and **cache** policies and/or stripe sizes for a particular controller?

A: All RAID controllers have their own default read, write, and cache policies and stripe sizes. It is possible that if you provide an invalid policy or a stripe size for a particular controller, RAIDCFG may not give an error but will and create the virtual disk with the default policy and/or stripe size.

Q: When I execute RAIDCFG with the RAID level of "01," RAIDCFG creates RAID 1, not RAID 01.

A: This behavior occurs because the RAIDCFG CLI parser ignores zeros preceding any parameter value. For example, `-r=01` will be parsed as `-r=1`.

Q: On PowerEdge 1655MC system running embedded Linux, RAIDCFG displays the error: `megadev: Bad file descriptor.`

A: Ignore this message. The error message is generated by Megalib because it is trying to open a device driver that is not present on your system.

Q: What is strict creation?

A: Strict creation is an optional flag that has been added to the **create virtual disk** command. It allows you to create virtual disks only if the array disks (that is, hard drives) are within a specified percentage of disk space of each other.

Q: Why is the drive location of array disks displayed like 0:0:0?

A: The nexus used to display array disks is "channel:target:lun." The Logical Unit Number (LUN) is always "0." The channel and the target IDs are necessary to identify array disks when dealing with SCSI devices.

When dealing with SAS devices, the array disk location is displayed as "channel:target:enclosure." It is possible to have the enclosure value as non-zero. If the enclosure has a non-zero value, then all three numbers (channel, target, and enclosure) have to be included on the command line.

Q: Do I need to specify RAID type and size when creating virtual disk(s)?

A: No. The default RAID type is RAID 0. If RAID size is not provided, the maximum size allowed for virtual disk is created. The mandatory fields required to create virtual disks are the controller slot ID and the array disks.

Q: What is a hotspare (failover drive)?

A: A hotspare is an extra and unused disk drive that is part of an array disk subsystem. A hotspare is always in standby mode. If a disk failure occurs, the hotspare replaces the failed drive without interrupting the system.

Q: What is **mr2kserv.exe**?

A: This is a service needed to configure all LSI RAID controllers in Windows PE. It provides Plug and Play support.

Q: What features have changed from the MS-DOS® version of RAIDCFG to the Windows PE and embedded Linux versions?

A: New features for the Windows PE embedded Linux RAIDCFG include:

Support for the following controllers:

- PERC 4/SC, PERC 4/DC, PERC 4/Di, and PERC 4/IM
- PERC 4e/DC, PERC 4e/Di, and PERC 4e/Si
- PERC 5/E Adapter, PERC 5/I Integrated, and PERC 5/i Adapter
- CERC SATA 1.5/6-Channel (CH) or CERC SATA 1.5/2S (These controllers are supported in Linux only.)
- SAS 5iR Integrated, SAS 5iR Adapter
- SAS 6iR Integrated, SAS 6iR Adapter
- PERC 6/E Adapter, PERC 6/i Integrated, PERC 6/i Adapter

Features that have been removed from the DOS version include:

- The **--name** and **--wait** options in the create command
- The **--runlocation** and the **--scratchlocation** options
- The **--getpercentcomplete** environment variables

Q: What is the minimum virtual disk size that I can create?

A: For CERC-SATA 1.5/6ch controllers:

- RAID 0: 10MB
- RAID 1: 9MB
- RAID 5: 9MB

For PERC 4 (except PERC 4/IM) controllers:

- RAID 0: 10MB
- RAID 1: 10MB
- RAID 5: 10MB

For PERC 5 controllers:

- RAID 0: 100MB
- RAID 1: 100MB
- RAID 5: 100MB

For PERC 6 controllers:

- RAID 0: 100MB
- RAID 1: 100MB
- RAID 5: 100MB
- RAID 6: 100MB
- RAID 60: 100MB

Q: Why can't I see any controllers using RAIDCFG?

A: Ensure that the RAID controller(s) can be seen in the BIOS during POST. If the BIOS does not detect the controller, then RAIDCFG will also not detect it. Also ensure that the controller is set to RAID mode and not any other mode in the controller BIOS.

Q: What is span length?

A: Span length is the field needed when you try to create a RAID 50. The number represents the span size of the RAID 5.

Q: What pre-operating system environments are CERC SATA 1.5/6ch, and CERC SATA 1.5/2s controllers supported on?

A: They are supported only on embedded Linux.

Q: PARTCFG cannot enumerate the virtual disk(s) created by RAIDCFG. What should be done?

A: A reboot is required for PARTCFG to enumerate virtual disks after creating a virtual disk using the RAIDCFG command.

SYSCFG Questions

NOTE: In a Linux environment, the DTK utilities do not have a **.exe** extension.

Q: On *x7xx* systems, when I set arguments for **--sata0** and **--sata1** using DTK CLI and reboot the system, the BIOS setup shows the earlier settings for **--sata0** and **--sata1**.

A: You need to set the **--idectrl** to auto before you can change the settings for **--sata0 and --sata1.**

Q: On *x7xx* systems, the BIOS setup shows **--conred** arguments of **--serial1** and **--rac**, whereas the DTK CLI shows them of **--serial1** and **--serial2**.

A: The **--serial2** argument in the DTK CLI corresponds to the **--rac** argument in BIOS setup.

Q: Can I use the **SYSCFG** utility to configure Point-to-Point Protocol (PPP) over the serial port to my access my RAC?

A: No. The baseboard management controller (BMC) firmware does not support PPP configurations for the serial port.

Q: The SYSCFG utility **--lancfgparams** option has suboptions that configure the IP address, Gateway, and Subnet Mask for the system BMC. There is also a MAC address. Are these the same as the managed system's operating system network parameters?

A: No. The BMC has its own IP address, Gateway, Subnet Mask, and MAC address values.

Q: What is the minimum configuration needed to support BMC Platform Event Filtering?

A: You must enable LAN-channel access "pefalerting" and ensure that you set the BMC IP address and Gateway values for alerting.

Q: I loaded the factory defaults for my system's BMC. Then, I enabled BMC user ID 3. What is the username for this user ID?

A: The default username for user IDs 3 to 10 is NULL.

Q: Can I use this NULL username to remotely log in to a system's BMC?

A: No. Remote login using a NULL username is not allowed. It is recommended that you always provide a non-null, valid username for the user ID when you enable a BMC user ID.

Q: Are duplicate usernames allowed in the BMC?

A: No. If you enter a username value that already exists, the **SYSCFG** utility returns an error code of 89, which means `This username is already in use. Enter a unique username`.

Q: In **--solcfgparams**, when I set the Serial Over LAN (SOL) character send threshold using solcharsendthreshold to be 225, I get a `Hardware subsystem error. Invalid data field in request`. What are the valid values?

A: The range of valid values for **--solcfgparams**, **--solcharsendthreshold** are from 1 to 220.

Q: How do I enable console redirection on a modular system?

A: Set the **--serialcomm** option value to **enableconred**. For example, `syscfg -serialcomm=enableconred`.

Embedded Linux Questions

Q: After installing Linux, I am not able to boot to the utility partition by pressing the **F10** key during reboot.

A: Recreate the utility partition using the **upinit.sh** script and install **grub** in the boot partition.

Q: When booting DTK through PXE, my system displays error messages, unknown behavior, and also leads to a kernel panic sometimes. However, booting from the CD causes no problems.

A: Ensure that you specify the correct RAMDISK size in the **pxe config** file before booting through PXE. (A sample **pxe config** file is provided in **/mnt/cdrom/isolinux/isolinux.cfg**.)

Q: What is the version of the Linux kernel used?

A: The Linux kernel is derived from 2.6.9-42. The source RPM (SRPM) for the Linux kernel is available on `/mnt/cdrom/srpms`.

Q: Can I use a 2.5 Linux kernel?

A: Yes. You could integrate the DTK tools into your own embedded Linux. See "Deployment Using Customized Embedded Linux."

Q: Can we use the same sample scripts that we have from earlier versions of DTK?

A: The scripts written for the previous versions (2.x) of DTK will not work in DTK 3.x versions for embedded Linux. See the directory **/opt/dell/toolkit/template/scripts** for the sample scripts.

Q: After I install a legacy operating system on a newer Dell system, the Linux kernel does not boot and panics.

A: After installing the initial ramdisk image, you could install the driver or you could update to a newer version of the kernel.

Q: After the DTK deployment is complete, the Red Hat® Enterprise Linux® Server installation keeps going into attended mode or halts with the message `ks.cfg not found`.

A: Always use **initrd** for the Red Hat Enterprise Linux Server installation from the Dell Systems Build and Update Utility available on the path **server_assistant/drivers** on the *Dell Systems Management Tools and Documentation* DVD.

If the **initrd** is not available for your operating system, try the latest update of Red Hat Enterprise Linux Server. Otherwise, you might need to prepare driver disks for your network or disk controller. Use the drivers from the Systems Build and Update Utility available on the path **server_assistant/drivers** on the *Dell Systems Management Tools and Documentation* DVD.

Windows PE Questions

Q: My system gets the error `Illegal or Missing File Types Specified in Section Files.SCSI.Name` during an unattended installation of a supported Windows operating system on supported Dell systems.

A: This behavior can occur when the line in the **Txtsetup.oem** file under the [Files.SCSI.*name*] heading is not a supported file type. To resolve this behavior, you must remove the line in the **Txtsetup.oem** file. The dynamic-link library (DLL) file can be copied to the correct location of the installation by placing it in the **I386\$Oem$\$$\OEMDIR** folder. (**OEMDIR** is the destination folder where the file would normally be located, if installed to a running operating system. For example, INF files are normally found in the **%SystemRoot%\INF** folder. The correct **OEMDIR** destination can be found

by searching the INF file used to install the device or driver.) For more information, see Microsoft Knowledge Base Article 275334 on the Microsoft website.

Q: When I try to install network interface controller (NIC) drivers into my Windows PE image using **DRVINST.EXE** [provided in the OEM Preinstallation Kit (OPK) CD], the **DRVINST.EXE** fails with an error message that there are some files missing.

A: Dell recommends that you use **DRIVERINST.BAT** (provided with DTK).

However, if you use **DRVINST.EXE** from Microsoft to install the NIC drivers into your Windows PE image, you need to download and extract required NIC drivers for Windows 2000 and Windows Server 2003 to a common directory. Then, merge the contents of `Base_drivers` from both. Use the merged contents with **DRVINST.EXE** to integrate the NIC drivers into your Windows PE image.

Q: In Windows, I see a blue screen when I delete a virtual disk and create a new one without initialization and attempt to format the disk.

A: Initialize (fast init) the drives before formatting the disk. This will prevent the issue.

Index

C

capturing configuration information
embedded Linux, 122
Windows PE, 66

capturing the system BIOS & BMC configuration
embedded Linux, 122
Windows PE, 66

CD-Based local deployment
embedded Linux, 85
Windows PE, 25

CD-Based local deployment, networking enabled
embedded Linux, 84
Windows PE, 26

choosing a deployment method
embedded Linux, 83
Windows PE, 25

configuration files, 11

configuring an optimally configured source system
embedded Linux, 104

contents
DTK embedded Linux, 95
DTK Windows PE, 34

creating a master file
master batch file in Windows PE, 55

creating deployment directory structure
embedded Linux, 100
Windows PE, 42

creating disk partitions
embedded Linux, 118
Windows PE, 62

D

deployment directory structure
embedded Linux, 101
overview, 31, 91
Windows PE, 43

deployment matrix
embedded Linux, 87
Windows PE, 28

deployment methods, 18

deployment overview
CD-Based, 22
network-based, 21

deployment prerequisites
embedded Linux, 17
Windows PE, 16

document organization, 12

drivers
embedded Linux, 99
Windows PE, 40

DTK, 95

embedded Linux contents, 95
installing, 91, 105
DTK deployment overview, 19
DTK prerequisites and requirements, 14

E

editing DTK task scripts
embedded Linux, 120
Windows PE, 64
editing the files
embedded Linux, 108
Windows PE, 55
embedded Linux contents
DTK, 95

I

installation prerequisites
embedded Linux, 15
Windows PE, 15
installing DTK, 91, 105
embedded Linux, 91

M

Master Batch File
sample, 54
what is, 54

O

operating system installation files
embedded Linux, 102
Windows PE, 45
overview
deploying, 21
preparing for deployment, 19

P

performing a scripted deployment
embedded Linux, 108
Windows PE, 55
preparing operating system installation scripts
embedded Linux, 125
Red Hat Enterprise Linux Server, 125
SUSE Linux Enterprise Server, 127
Windows PE, 69
prerequisites for deployment
embedded Linux, 17
Windows PE, 16
prerequisites for installation
embedded Linux, 15
Windows PE, 15

R

RAC

configuring, 51
Running, 135
running a scripted deployment
embedded Linux, 135
Windows PE 2005, 73
running a scripted deployment, embedded Linux
network-based deployment, 135
third-party deployment solution framework, 134
using customized embedded Linux, 134
using Dell-provided embedded Linux, 131
running a scripted deployment, using Dell-provided embedded Linux
CD-based deployment, 132
Network-based deployment, 131
running a scripted deployment, Windows PE
burning ISO image on a CD, 76, 79
CD-Based local deployment, 77, 79
CD-Based local deployment, networking enabled, 77, 80
copying tools from OPK, 73
creating a standard Windows PE, 74
creating bootable Windows PE ISO image, 76
integrating DTK directory structure, 75
third-party deployment solution framework, 77, 80
Running Update Packages
customized embedded linux, 135
Dell-provided embedded linux, 135

S

sample configuration files
embedded Linux, 97
sample script files
master deployment file for embedded Linux, 115
sample scripts
embedded Linux, 98
Windows PE, 38
scripted deployment in embedded Linux
autoinst.xml, 129
capturing the RAC configuration, 123
ks.cfg, 126
lininst.sh, 111, 125
partcfg.sh, 122
raccap.sh, 123
racrep.sh, 120
raidcfg.sh, 113, 121
suseinst.sh, 127
sysrep.sh, 121
upinit.sh, 115, 124
scripted deployment in Windows PE

capturing the RAC configuration, 67
PARTCFG.BAT, 66
RACCAP.BAT, 67
RACREP.BAT, 64
RAIDCFG.BAT, 65
SYSCAP.BAT, 66
SYSREP.BAT, 57, 65
TKENVSET.BAT, 58
UNATTEND.TXT, 70
UPINIT.BAT, 68
W2K8INST.BAT, 71
WININST.BAT, 69

specifying DTK installation paths
embedded Linux, 116
Windows PE, 60

supported operating systems
deploy embedded Linux, 18
deploy Windows PE, 18
to deploy, 18
to run DTK utilities, embedded Linux, 15
to run DTK utilities, Windows PE, 15

supported operating systems to run DTK
embedded Linux, 15
Windows PE, 15

supported systems, 15

supporting documentation
other dell documents you might need, 23
other documents you might need, 24

supporting tools and system files
embedded Linux, 99
Windows PE, 41

system utilities and supporting files
embedded Linux, 95
Windows PE, 34

system-specific deployment files
embedded Linux, 101
Windows PE, 43

U

upgrading
embedded Linux, 91
Windows PE, 31

using the DTK sample scripts
embedded Linux, 107
preparing for a full deployment, 53, 107
Windows PE, 54

utilities and files needed for deployment
embedded Linux, 101
Windows PE, 43

W

what's new for this release, 14

Windows PE contents
DTK, 34