Tru64 UNIX and TruCluster Software Products

Patch Kit Installation Instructions

June 2005

Product Version: HP Tru64 UNIX Version 5.1B-2 and higher patch kits

This guide provides instructions for installing, removing, and working with Release patch kits, Customer-Specific patch kits (CSPs), and Early Release patch kits (ERPs) using the `dupatch` utility, which is included with HP Tru64 UNIX patch kits.

The information in this guide provides examples and descriptions for the Inclusive patch kit structure, which was introduced with Version 5.1B-2. If you are working with earlier patch kits, we recommend using the instructions that came with your patch kit. We also provide a guide named *Installation Instructions for Pre-Inclusive Patch Kits* on the patch documentation Web. See Patch Process Resources for information.

For information about individual patches and information that is specific to a patch kit, see the *Patch Summary and Release Notes* document for the kit you are installing.

Hewlett-Packard Company
Palo Alto, California

Contents

About This Manual

1 Patch Process Overview

1.1 Changes to the dupatch Utility 1-1
1.2 Using dupatch 1-1
1.2.1 dupatch Overview 1-1
1.2.2 Invoking dupatch and Installing New Patch Tools 1-2
1.3 Patch Applicability 1-3
1.4 Patch Reversibility 1-3
1.5 Using the Patch Tracking Menu 1-4
1.6 Using the Patch Documentation Menu 1-4
1.7 Version Switches 1-6
1.8 General Issues and Restrictions 1-6
1.8.1 When Single-User Mode Is Recommended 1-6
1.8.2 Use Clean Directory for Each Patch Kit 1-7
1.8.3 Patching a System Prior to Creating a Cluster 1-7
1.8.4 RIS and DMS Unsupported for Patch Installation 1-7
1.8.5 Direct setld Installation and Removal of Patch Subsets Is Not Allowed 1-7
1.8.6 Limitation for /var/adm/patch/backup Directory Handling 1-7
1.8.7 Do Not Enter Ctrl/c During Installation Phase 1-7
1.8.8 Removing Patches Containing Customized Files 1-7
1.8.9 Release Patches Do Not Automatically Supersede CSPs 1-8
1.8.10 Impact on System Upgrades to Later Versions of Tru64 UNIX ... 1-8

2 Preparing for the Installation

2.1 Performing a Patch Preinstallation Check 2-1
2.2 Creating a Baseline 2-3
2.2.1 Phase 1 – System Evaluation 2-4
2.2.2 Phase 2 – Patch Layered Product Conflicts 2-5
2.2.3 Phase 3 – Identifying Manually Installed Patches 2-5
2.2.4 Phase 4 – Handling Missing or Unknown Files on Your System . 2-5
2.2.4.1 Manually Installed CSPs 2-5
2.2.4.2 Manually Installed Release Patches 2-6
2.2.4.3 User Customized Commands and Utilities 2-6
2.2.5 Phase 5 – Enabling dupatch to Overwrite Changed System Files 2-6
2.2.6 Phase 6 – Report CSPs with Inventory Conflicts 2-7
2.2.7 Phase 7 – Enable patches with File Applicability Conflicts 2-7
2.2.8 Steps for Running the Baseline Procedure 2-7

3 Patch Installation and Removal Instructions

3.1 Before You Begin the Installation 3-1
3.2 Expanding the Patch Kit Tar File 3-1
3.3 Choosing Single-User or Multiuser Mode 3-2
3.3.1 Installing Patches from Single-User Mode 3-2
3.3.2 Installing Patches from Multiuser Mode 3-3

3.4 Common Installation Steps 3-4
3.5 Rebuilding the Kernel 3-5
3.6 Rebooting the System 3-7
3.6.1 In Single-User Mode 3-7
3.6.2 In Multiuser Mode 3-7
3.7 Post-Installation Actions 3-7
3.7.1 Enabling the Version Switch After Installing a New Style Patch Kit 3-7
3.7.2 Remove Temporary Directory 3-8
3.7.3 Adding the Worldwide Language Support 3-8
3.8 Removing Patches 3-8
3.8.1 Overview 3-8
3.8.2 Important Tasks Required Before Removing Patches and Rebooting System 3-9
3.8.2.1 Run Mandatory Script Before Removing New Style Patch Kits 3-9
3.8.2.2 Changes to System May Need to Be Reversed 3-10
3.8.2.3 Script Must Be Run Prior to Reboot on Certain Version 5.1B Systems 3-10
3.8.3 Running dupatch to Remove Patches 3-11

4 Rolling Upgrade

4.1 Rolling Upgrade Supported Tasks 4-2
4.2 Unsupported Tasks 4-4
4.3 Rolling Upgrade Procedure 4-5
4.4 Removing Patches Installed During a Rolling Upgrade 4-10
4.4.1 Caution on Removing Version Switched Patches 4-10
4.4.2 Steps Prior to the Switch Stage 4-10
4.4.3 Steps for After the Switch Stage 4-11
4.5 Displaying the Status of a Rolling Upgrade 4-11
4.6 Undoing a Stage 4-12
4.7 Rolling Upgrade Commands 4-13
4.8 Rolling Upgrade Stages 4-15
4.8.1 Preparation Stage 4-16
4.8.2 Setup Stage 4-18
4.8.3 Preinstall Stage 4-19
4.8.4 Install Stage 4-19
4.8.5 Postinstall Stage 4-20
4.8.6 Roll Stage 4-20
4.8.7 Switch Stage 4-21
4.8.8 Clean Stage 4-21
4.9 Tagged Files 4-22
4.10 Version Switch 4-24
4.11 Rolling Upgrade and Layered Products 4-24
4.11.1 General Guidelines 4-25
4.11.2 Blocking Layered Products 4-25
4.12 Rolling Upgrade and RIS 4-26

5 No-Roll Patching

5.1 Overview 5-1
5.2 Steps for Running a No-Roll Procedure 5-2
5.3 Throwing the Version Switch 5-3

5.4 Removing Patches 5-3

A Viewing Log files

B Common Error, Warning, and Informational Messages

B.1 Patch Preinstallation Check and Installation Messages B-1
B.1.1 Patch Installation Blocked by Unknown System File B-1
B.1.2 Patch Installation Blocked by Missing System File B-2
B.1.3 Installation Blocked by Layered Product Collision B-2
B.1.4 Patch Installation Blocked by Dependencies on Other Patches .. B-3
B.1.5 Patch Installation Blocked by Missing Product Subset B-3
B.1.6 Patch Installation Blocked by Disk Space B-4
B.1.7 Patch Installation Blocked by Installed Patch or Subset B-4
B.1.8 Patch Installation Blocked by an Existing CSP B-5
B.1.9 The dupatch Tools Are Outdated B-5
B.1.10 Some Patches Must Be Made Reversible B-6
B.2 Patch Removal Messages B-6
B.2.1 Patch Removal Blocked by Missing Patch Backup Files B-6
B.2.2 Patch Removal Blocked by Dependencies on Other Patches B-6
B.2.3 No Original Files Restored When Patch Is Removed B-7
B.3 TruCluster Specific dupatch Messages B-7
B.3.1 System Not Adequately Prepared B-7
B.3.2 Rolling Upgrade in Progress (Installation) B-7
B.3.3 Rolling Upgrade in Progress (Baselining) B-8
B.3.4 Version 5.0 Wave 4 Cluster is Unsupported B-8
B.3.5 Patch Removal Fails Because Needed File Is Unavailable B-8
B.3.6 Patch Removal Fails Because of a Version Switch B-8
B.3.7 dupatch Cannot Create Needed File B-9
B.3.8 Insufficient Free Space (File System Full) B-9

C Using dupatch from the Command Line

C.1 Installing and Removing Release Patch Kits C-1
C.2 Deleting a CSP C-1
C.3 dupatch Reference Page C-2
dupatch(8) C-3

D Inclusive Patch Kits

Glossary

Index

Examples

A-1 Sample Event Log A-2

Figures

4-1 Rolling Upgrade Flow Chart 4-2

Tables

4-1	Rolling Upgrade Tasks Supported by Version 5.1A and Version 5.1B	4-3
4-2	Time Estimates for Rolling Upgrade Stages	4-5
4-3	Undoing a Stage	4-12
4-4	Stages and clu_upgrade Versions When Performing a Rolling Upgrade from Version 5.1A	4-14
4-5	Stages and clu_upgrade Versions When Performing a Rolling Upgrade from Version 5.1B	4-14
4-6	Blocking Layered Products	4-25

About This Manual

This manual provides instructions for installing and removing patches that are provided by Hewlett-Packard Company in its Tru64 UNIX and TruCluster software products patch kits. It also describes baselining techniques and provides other information for working with patches.

The information in this guide provides examples and descriptions for the Inclusive patch kit structure, which was introduced with Version 5.1B-2. If you are working with earlier patch kits, we recommend using the instructions that came with your patch kit. We also provide a guide named *Installation Instructions for Pre-Inclusive Patch Kits* on the patch documentation Web site. See *Patch Process Resources* for information.

Audience

This manual is for those who install and remove patch kits and manage patches after they are installed.

Changes to This Manual

The revision of the `dupatch` utility that is installed with Tru64 UNIX Version 5.1B-3 makes it easier to use with the Inclusive-style patch kits that were introduced in Version 5.1B-2.

Although you can use this `dupatch` version to install and work with the old-style patch kits, the focus of this manual is on the interface for the inclusive patch kits. You can see explanations and examples for working with old-style kits by using the *Patch Kit Installation Instructions* manual that shipped with an earlier patch kit. You can find a previous manual on the Patch Web site. See *Patch Process Resources* for more information.

In addition to describing the new `dupatch` features, the following changes are among those included in this manual:

- This manual has been reorganized. For example, information about the baselining process is now provided in one place.
- New examples are provided throughout the manual.

See *Section 1.1* for an overview of the changes to the `dupatch` utility.

Organization

This manual is organized as follows:

Chapter 1	Introduces the `dupatch` utility and describes its features.
Chapter 2	Provides information to be aware of when installing and removing patches.
Chapter 3	Describes the procedures for installing and removing patches.
Chapter 4	Describes the rolling upgrade process for patching a system running TruCluster Server Version 5.0A or higher while the cluster is in operation. This process is also used for upgrading to a new version of the TruCluster software software or for doing an upgrade and a patch together.

Chapter 5	Describes the no-roll patch process, which provides a way to apply patches to a cluster quickly in order to minimize downtime and reduce the number of reboots required.
Appendix A	Helps you understand the log files generated by `dupatch`.
Appendix B	Describes error messages you might see while installing, removing, or maintaining patches.
Appendix C	Provides information about using the `dupatch` command-line interface and documents the `dupatch`(8) reference page.

Related Documentation

In addition to this manual, the following documentation may be helpful in the patching process:

- The *Patch Summary and Release Notes* for the patch kit you are working with.
- *Technical Updates for Tru64 UNIX Version 5.0 and Higher Patch Kits* or *Technical Updates for Tru64 UNIX Versions 4.0F and 4.0G*, which report any information about restrictions and problems that may have been discovered since the release of these patch kits.
- *Patching Best Practice*
- Tru64 UNIX *Installation Guide*
- Tru64 UNIX *System Administration*
- TruCluster Server *Cluster Installation*
- TruCluster Server *Cluster Administration*

See *Patch Process Resources* for Web sites where you can find this documentation.

Patch Process Resources

We provide Web sites to help you with the patching process:

- To obtain the latest patch kit for your operating system and cluster go to:

  `http://www.itrc.hp.com/service/patch/mainPage.do`

- To obtain early release patches (ERPs):

  `http://h30097.www3.hp.com/unix/EarlyReleasePatch-download.html`

- To view or print patch-related documentation go to:

  `http://h30097.www3.hp.com/docs/patch/`

  Here you can find patch-specific technical updates, release notes for current and previous patch kits, this installation guide, and other information that can help you with the patching process.

- To view or print patch-related documentation go to:

  `http://h30097.www3.hp.com/docs/`

  Here you can find Tru64 UNIX documentation, TruCluster software product documentation, operating system and other technical updates, and other information to help you with your Tru64 UNIX systems.

- To visit the Tru64 UNIX homepage go to:

  `http://h30097.www3.hp.com/`

- To visit our main support HP page go to:

  `http://h71025.www7.hp.com/support/home/index.asp`

Reader's Comments

HP welcomes any comments and suggestions you have on this and other Tru64 UNIX manuals.

You can send your comments in the following ways:

- Fax: 603-884-0120 Attn: UBPG Publications, ZKO3-3/Y32
- Mail:

  Hewlett-Packard Company
  HCTO Information Development Manager
  ZK03-3/Y32
  110 Spit Brook Road
  Nashua, NH 03062-9987

Please include the following information along with your comments:

- The full title of the manual.
- The section numbers and page numbers of the information on which you are commenting.
- The version of Tru64 UNIX that you are using.
- If known, the type of processor that is running the Tru64 UNIX software.

The Tru64 UNIX Publications group cannot respond to system problems or technical support inquiries. Please address technical questions to your local system vendor or to the appropriate HP technical support office. Information provided with the software media explains how to send problem reports to HP.

Conventions

This guide uses the following conventions:

file	Italic (slanted) type indicates variable values, placeholders, and function argument names.
Ctrl/*x*	This symbol indicates that you hold down the first named key while pressing the key or mouse button that follows the slash. In examples, this key combination is enclosed in a box (for example, Ctrl/c).
# **setld**	Boldface type in interactive examples indicates typed user input.
device names	Operating system versions before Version 5.0 use different names than those of Version 5.0 and higher. In general, this manual uses the Version 5.0 names. For example, where a partition name is represented by `/dev/disk/dsk3g`, the Version 4.0n name might be `/dev/rz3g`.
⋮	The vertical ellipsis is used in output examples to replace redundant information. This is information you would see on your terminal screen or in a log file created by `dupatch`, but is not particularly useful in the examples in this manual.
Glossary Terms	In the online version of this document, various terms are linked to the *Glossary*. By clicking on the term, you will be

taken to its definition. You can return to the place you were reading by clicking on your browser's Back button.

1

Patch Process Overview

This chapter introduces you to the dupatch utility for installing, removing, and managing patches. See Chapter 3 for instructions on installing and removing patches from the Tru64 UNIX operating system and the TruCluster software products.

1.1 Changes to the dupatch Utility

Beginning with Version 5.1B-2, HP changed the way Tru64 UNIX patch kits are installed by introducing the concept of Inclusive Patch Kits. If you did not install Version 5.1B-2 but have installed earlier kits, you may want to review an overview of the installation changes as described in Appendix D.

Version 5.1B-3 introduced several new changes to the way the dupatch utility installs, removes, and works with patches:

- Before you can install this kit you must accept the conditions included in the license that the dupatch utility displays (see Section 3.4). You can read this license in the *Patch Summary and Release Notes* for Version 5.1B-3.
- You can now delete the patch kit by kit name rather than by specifying individual patches. With the introduction of this feature, you can easily delete patches interactively using the dupatch graphical interface or from the dupatch command line. This feature also works on on pre-V5.1B-3 kits. See Section 3.8.3.
- You can delete patches in multiuser mode. See Section 3.8.1.
- You can force the installation of the patch kit even if file conflicts exist. This feature is an extension of the dupatch baselining feature. See Section 2.2.
- A new command-line option — `dupatch -track -type patch_level` — provides a single command that lists a full description of the patch kits, CSPs, and ERPs installed on your system. See Appendix C.

1.2 Using dupatch

The dupatch utility is provided as an interactive and command-line tool for working with Patch kits. The following sections provide an overview of dupatch and describe the procedure for installing the most current version.

1.2.1 dupatch Overview

All Tru64 UNIX and TruCluster software Release Patch Kits are installed, removed, and managed using the setld-based dupatch utility, which provides you with menus that step you though the various tasks.

The dupatch utility is also used for installing many of the Customer-Specific Patch Kits (CSPs) , and Early Release Patch Kits (ERPs) . Although the examples and descriptions provided in this manual, in general, refer to Release Patch Kits, the information is similar for CSPs and ERPs that install using dupatch.

The dupatch utility is interactive, but you can also run it from the command line using command options. For information about using the command-line interface, see Appendix C, which includes the dupatch(8) reference page.

For clustered systems running TruCluster dupatch is run in conjunction with the rolling upgrade (see Chapter 4) or no-roll (see Chapter 5) procedures.

With dupatch, you can perform the following actions:

- Install and remove patches.
- View patch tracking and management information.
- Track current dupatch-installed patches and Customer-Specific patches.
- Establish a baseline for systems that had manually installed system files placed on them.
- Ensure the correct handling of customized system configuration files so that customizations are not lost (for example, conf.c). These files are also referred to as system-protected files (.new..).
- Validate patch applicability to existing system files (collision detection).
- View the patch-specific documentation.

Because dupatch manages patch interdependencies, direct setld installations (setld -l) and deinstallations (setld -d) are disabled.

Most dupatch operations generate log files that record the step-by-step procedures performed during the operation. For information about log files see Appendix A.

1.2.2 Invoking dupatch and Installing New Patch Tools

After you have made the patch kits available to the system being patched, run dupatch from root or you can change directories to patch_kit, which contains the dupatch utility:

From root:

```
# /patches/pk4/patch_kit/dupatch
```

From the patch_kit directory:

```
# cd /patches/pk4/patch_kit
# ./dupatch
```

If new patch tools are available they will be loaded and you will see messages similar to the following:

```
* A new version of patch tools required for patch management
  is now being installed on your system.

* Tools updated, invoking the updated Patch Utility...
```

The dupatch utility saves information on the tools that have been loaded to the log file /var/adm/patch/log/Dupatch_load_*date*.log. (See Appendix A for information about log files.)

Note

To install the latest version of the patch tools, it is important that you run the dupatch utility located in the /patch_kit directory every time you obtain a new patch tar file or a new Tru64 UNIX Patch CD–ROM. See Section C.1 for information you need to be aware of when installing from the command line.

After the new tools have been loaded, dupatch prompts you for the path to the patch kit files. After you specify the path (or press Return if the patch kit is in your current directory) you will see the main menu. For example:

```
Enter path to the top of the patch distribution,
or enter "q" to get back to the menu :  /patches/pk4/patch_kit
```

```
Tru64 UNIX Patch Utility (Rev. 48-00)
=========================
 - This dupatch session is logged in /var/adm/patch/log/session.log

    Main Menu:
    ---------

    1)  Patch Kit Installation
    2)  Patch Kit Deletion
    3)  Patch Kit Documentation

    4)  Patch Tracking
    5)  Patch Baseline Analysis/Adjustment

    h)  Help on Command Line Interface

    q)  Quit

Enter your choice:
```

1.3 Patch Applicability

Patch applicability to the existing system files is done on a file-by-file basis for each patch. This ensures that the installation of a patch will not degrade or crash the system. The installation of a patch is blocked if any system files to be replaced by a patch are not valid predecessors of the patch files.

Patch applicability also enables consistency checking and reporting for the installation of Tru64 UNIX and TruCluster software patches.

In cases where a patch is blocked, informative messages are provided to assist you in determining how to proceed. Appendix B lists common error messages and suggested corrective actions.

The installation of a patch is blocked if any of the following conditions exist:

- The underlying software product subset is not installed — for example, if the applicable Tru64 UNIX or TruCluster software release subset is not installed.
- The `setld` inventory is inconsistent with the existing system files. This occurs when an operating system or TruCluster software `setld` subset is installed and individual operating system files that are part of that subset are moved, deleted, or replaced.
- The patch installation is blocked if any existing system files (files that are targeted to be updated by a patch) have changed and cannot be related to previous versions of the patch. This ensures that operating system files that change due to other explicit system administrator action (for example, layered product patches or non-`dupatch` installed CSP installations) are not inadvertently overwritten. You must take special action, through the baseline feature to enable patch installation in this situation.

1.4 Patch Reversibility

By default, Release Patch Kits are made reversible during the installation so you revert your system to its state prior to the installation. If you choose to make patch kits nonreversible, you will not be able to uninstall the kit.

Customer-Specific patch kits are forced to be reversible when the CSP kit is manufactured. This forced reversibility overrides the reversibility option provided by `dupatch` during installation.

Patch reversibility is dependent upon saving the existing system files that will be updated by the patch. Saving these files requires the availability of adequate storage space in `/var/adm/patch/backup`, which can be a mount point for a separate disk partition, an NFS mount point, or a symbolic link to another file

system. This allows you to configure your system to reduce the impact on system disk space for the /, /usr, and /var partitions.

The dupatch utility checks for the required storage space prior to patch installation. Patch installation is prevented if adequate backup space is unavailable.

Section 2.1 includes an example of the dupatch output regarding patch reversibility.

1.5 Using the Patch Tracking Menu

The dupatch patch-tracking capability lets you view information about installed patches, such as lists of release patches, CSPs, and ERPs installed on the system and which patch kits you have installed.

For example, the following dupatch output shows the patch tracking menu with the List Installed patches menu item selected:

```
    Patch Tracking Menu:
    --------------------
    1)  List installed patches
    2)  List installed patch files
    3)  List patch kit information for installed patches
    4)  Show Patch History for selected patches
    5)  Show System Patch History

    b)  Back to Main Menu
    q)  Quit  Enter your choice: 1

    Patch Tracking Selection Menu:
    ------------------------------

    1)  List Release Patches
    2)  List Customer Specific Patches
    3)  List All Patches

    b)  Back to Tracking Menu
    q)  Quit

Enter your choice:

Gathering details of relevant patches, this may take a bit of time
 Patches installed on the system:
 -------------------------------
  (depending upon the number of patches you installed, this may take awhile)

 - Tru64_UNIX_V5.1B / Commands, Shells, & Utilities Patches:
        Patch 25022.00 - SP04 OSFDCMT540
        Patch 25080.00 - SP04 OSFTCLBASE540
        Patch 26022.00 - SP05 OSFDCMT540
        Patch 26080.00 - SP05 OSFTCLBASE540

 - Tru64_UNIX_V5.1B / Common Desktop Environment (CDE) Patches:
        Patch 25015.00 - SP04 OSFCDEDT540 (SSRT2405)
        Patch 25016.00 - SP04 OSFCDEMAIL540
.
.
.
        Patch 26085.00 - SP05 OSFX11540 (SSRT4831 SSRT4802 SSRT4800 SSRT4721)
        Patch 26086.00 - SP05 OSFXADMIN540

Press RETURN to get back to the Patch Tracking Menu...
```

1.6 Using the Patch Documentation Menu

When you select the Patch Documentation item of the main menu, dupatch returns a menu that gives you access to different information:

- Problem summaries

  Provide brief descriptions of the problems corrected by the patches. You can view the problems corrected by installed patches or by patches available from a specific kit.

- Full descriptions

  Provide complete descriptions of the problems corrected by the individual patches. You can view the problem descriptions for installed patches or for patches available from a specific kit.

- Special Instructions

  These files describe special instructions you need to be aware of for individual patches. You can view the instructions for installed patches or for patches available from a specific kit.

- Report identifiers

- Revision control strings

The following output shows the Patch Documentation menu and a typical session:

```
    Patch Documentation Menu:
    -------------------------

     Installed patches on the system
    1)  View problem summaries
    2)  View full descriptions
    3)  View special instructions
    4)  View Problem Report Identifiers
    5)  View Revision Control Strings
     Patches in the patch kit
    6)  View problem summaries
    7)  View full descriptions
    8)  View special instructions
    9)  View Problem Report Identifiers
   10)  View Revision Control Strings
     All (installed and non-installed) patches
   11)  View patch problem summaries
   12)  View patch full descriptions
   13)  View patch special instructions
   14)  View Problem Report Identifiers
   15)  View Revision Control Strings

    b)  Back to Main Menu
    q)  Quit

Enter your choice: 1

    Patch Documentation Selection Menu:
    -----------------------------------

    1)  List Release problem summaries
    2)  List Customer Specific problem summaries
    3)  List All problem summaries

    b)  Back to Documentation Menu
    q)  Quit

Enter your choice: 3
    There may be more patches than can be presented on a single
     screen. If this is the case, you can choose patches screen by screen
     or all at once on the last screen. All of the choices you make will
     be collected for your confirmation before any patches are examined.

 - Tru64_UNIX_V5.1B / Commands, Shells, & Utilities Patches:
     1) Patch 25022.00 - SP04 OSFDCMT540
     2) Patch 25080.00 - SP04 OSFTCLBASE540
     3) Patch 26022.00 - SP05 OSFDCMT540
     4) Patch 26080.00 - SP05 OSFTCLBASE540

 - Tru64_UNIX_V5.1B / Common Desktop Environment (CDE) Patches:
     5) Patch 25015.00 - SP04 OSFCDEDT540 (SSRT2405)
     6) Patch 25016.00 - SP04 OSFCDEMAIL540

.
.
.

- Tru64_UNIX_V5.1B / X11 Patches:
    49) Patch 25075.00 - SP04 OSFSER540
    50) Patch 25085.00 - SP04 OSFX11540

Or you may choose one of the following options:

    55) ALL of the above
```

```
    56) CANCEL selections and redisplay menus
    57) EXIT without examining any patches

Enter your choices or press RETURN to redisplay menus.

Choices (for example, 1 2 4-6): 1-4 7

Enter the output filename for the problem summaries for
installed patches, or < Return> to continue
(output to screen):

=======================================================

 Tru64_UNIX_V5.1B / Commands, Shells, & Utilities Patches:
        Patch 25022.00 - SP04 OSFDCMT540

A potential security vulnerability has been discovered,
where under certain circumstances, system integrity may
be compromised. This may be in the form of improper file
or privilege management. HP has corrected this potential
vulnerability
.
.
.

Press RETURN to proceed...

    Patch Documentation Selection Menu:
    -----------------------------------

    1)  List Release problem summaries
    2)  List Customer Specific problem summaries
    3)  List All problem summaries

    b)  Back to Documentation Menu
    q)  Quit

Enter your choice: q
```

The patch description information and special instructions are conveniently organized in the *Patch Summary and Release Notes* document that is packaged with each kit.

1.7 Version Switches

A version switch manages the transition of the active version to the new version of an operating system. The active version is the one that is currently in use.

With the Inclusive patch kits, you must manually enable the version switch. See Section 3.7.1 for more information

In the old-style patch kits, version switches are controlled by the `clu_upgrade -switch` command during a rolling patch. See Section 4.10 for more information.

1.8 General Issues and Restrictions

This section provides information you must be aware of when installing or removing patches. Be sure to check the *Patch Summary and Release Notes* document of the kit you are installing for any issues and restrictions that pertain to that installation.

1.8.1 When Single-User Mode Is Recommended

Although you can install patches in multiuser mode, we recommend that you bring down your system to single-user mode when installing patches that affect the operation of the Tru64 UNIX operating system or the product you are patching. If your system must remain in multiuser mode, apply the patches when the system is as lightly loaded as possible.

There are no restrictions on performing patch selection and preinstallation checking in multiuser mode. Patch removals can only be done in single-user mode.

1.8.2 Use Clean Directory for Each Patch Kit

When installing a patch kit downloaded from the Web, untar the file in a clean directory; that is, one that does not contain files from a previous patch kit. A failure to do this can have adverse consequences when installing the new kit.

1.8.3 Patching a System Prior to Creating a Cluster

Patching your system before creating your cluster can save you time, although if you do so, be aware that you cannot then remove the patch kit.

The following steps describe how to patch your system before creating a cluster:

1. Install and configure the Tru64 UNIX operating system.
2. Use the `setld` command to install the TruCluster software kit. If the TruCluster software kit is not loaded before the patch operation, patches for TruCluster software will not be loaded.
3. Patch the system.
4. Use the `clu_create` command to create the single-member cluster.

See the Tru64 UNIX *Installation Guide* for information about installing the operating system and the TruCluster *Cluster Installation* manual for information about creating your cluster.

1.8.4 RIS and DMS Unsupported for Patch Installation

Remote Installation Services (RIS) and Dataless Management Services (DMS) installations of patches are not supported. However, the patch kit installation mechanism does support network installation via NFS.

1.8.5 Direct setld Installation and Removal of Patch Subsets Is Not Allowed

You can install and remove Tru64 UNIX and TruCluster software patches only through `dupatch`. You cannot directly install or reinstall the patch subsets with `setld`. This ensures that patch tracking and management are not compromised.

1.8.6 Limitation for /var/adm/patch/backup Directory Handling

The patch management utility assumes there is one `/var/adm/patch/backup` directory per system. It does not handle placement of archived original files for multiple systems in one directory.

1.8.7 Do Not Enter Ctrl/c During Installation Phase

Do not enter a Ctrl/c command during the installation phase of the patch kit.

Caution

As with any system update, entering a Ctrl/c during this phase could leave the operating system software environment in an inconsistent and nonrecoverable state.

1.8.8 Removing Patches Containing Customized Files

If you use `dupatch` to remove a patch containing a customized file, messages similar to the following may appear in the session log file, `/var/adm/patch/log/session.log`:

```
- Tru64_UNIX_V5.1B / Network Patches:
        Patch 25020.00 - SP04 OSFCLINET540 (SSRT3653 SSRT2384 SSRT2275 ...)

        Customization found in ./etc/inetd.conf.

        Before the backup was restored, we had saved a copy of this file in:

                ./etc/inetd.conf.PreDel_OSFPAT02502000540

        Please compare ./etc/inetd.conf with this saved copy.

        If there are extra customizations you want to keep, you would need
        to merge them into ./etc/inetd.conf manually.

        ./etc/inetd.conf.PreDel_OSFPAT02502000540
        can be removed afterwards.
```

This message warns you to examine the removed patch for any customized files it may contain, which in this example is the file `/etc/inetd.conf`. In order to keep those customizations, you will have to manually add them.

The following are examples of such customized files:

- `/usr/var/spool/cron/crontabs/root`
- `/etc/sysconfigtab`
- `/usr/var/adm/sendmail/sendmail.cf`

1.8.9 Release Patches Do Not Automatically Supersede CSPs

Release patches do not automatically supersede `dupatch`-based Customer-Specific patches (CSPs). Any Release patch blocked by a CSP will result in a `dupatch` message. See Section B.1.7 for more information. See the release notes of the new style patch kits for a list of CSPs that are included in those patch kits. The *Patch Summary and Release Notes* document included with Version 5.1B-2 and higher includes a list of CSPs that were reconciled in the patch kit.

1.8.10 Impact on System Upgrades to Later Versions of Tru64 UNIX

In the presence of patches of layered products, certain procedures used to upgrade a system to a later version of Tru64 UNIX can lead to inconsistencies among operating system and layered product objects.

Note

After successfully installing a new version of Tru64 UNIX, you should obtain and install the latest patch kit that is applicable to that version.

2

Preparing for the Installation

This chapter describes information you need to be aware of before you install a patch kit. It also describes the steps to take for tasks such as performing a preinstallation check and a baselining operation.

2.1 Performing a Patch Preinstallation Check

To minimize system down time, you can perform the preinstallation check on a system running in multiuser mode, even if you will perform the actual installation in single-user mode.

The example in this section shows a preinstallation check that results in a patch that fails the check and is prevented from being installed. If this occurs, you would set the system patch baseline, as described in Section 2.2. If patches are prevented from being installed because dependent patches were not selected, choose the `select patches again` item and add the required patches that are missing.

If no patches are blocked, you can proceed to the installation phase, as described in Chapter 3.

Note that the menu you see will differ slightly, depending upon whether you log in from a pseudo-terminal or a system console. The following steps assume you logged in from a pseudo-terminal.

1. Log in as root.
2. From the main `dupatch` menu, enter `1` at the `Enter your choice` prompt:

```
Tru64 UNIX Patch Utility (Rev. 48-00)
==========================
 - This dupatch session is logged in /var/adm/patch/log/session.log

    Main Menu:
    ---------

    1)  Patch Kit Installation
    2)  Patch Kit Deletion
    3)  Patch Kit Documentation

    4)  Patch Tracking
    5)  Patch Baseline Analysis/Adjustment

    h)  Help on Command Line Interface

    q)  Quit

Enter your choice: 1
```

3. The program responds with the Patch Installation Menu. Enter `1` at the `Enter your choice` prompt:

```
             Patch Installation Menu:
            ------------------------

            1)  Pre-Installation Check ONLY
            2)  Check & Install in single-user mode w/ network services
            3)  Check and Install in Multi-User mode

            b) Back to Main Menu
            q) Quit

         Enter your choice: 1

Enter path to the top of the patch distribution,
or enter "q" to get back to the menu : ./patch_kit
```

```
    Tru64 Unix License Agreement
.
.
.

    To read the license again, type 'license'.
    Do you accept the license agreement? (y/n) : y

    Checking patch kit for transmission errors during download...

    Finished Checking patch kit checksums

    Gathering patch information...
      (depending upon the size of the patch kit, this may take awhile)

                   ***  Start of Special Instructions  ***
.
.
.

                   ***  End of Special Instructions  ***

     Press RETURN to proceed...
```

4. You have the option to make the patches reversible so you can revert the system to its state prior to the installation of a patch. The `dupatch` utility lists the following information. Press Return at the prompt to make the patches reversible. This is the recommended action.

```
        ------------------------------------------------------------------------
        To Make Patches Reversible - PLEASE READ THE FOLLOWING INFORMATION:

      - You have the option to make the patches reversible so you can
        revert the system to its state prior to the installation of a patch.

      - Reversibility is achieved by compressing and saving a copy of the
        files being replaced by the patches. These files would be restored
        to the system if you choose to delete a patch.

      - If you choose to make patches NON-reversible, then the system cannot
        be restored to the state prior to the installation of a patch; you
        will not be able to delete the patches later.

      - This patch kit may force a small set of patches to be reversible to
        ensure your upgrades to future versions of Tru64 UNIX are successful.
        The Patch Utility will make those patches reversible automatically.

        Refer to the Release Notes / Installation Instructions provided with
        this patch kit.

    Do you want the patches to be reversible? [y]: y

        By default, the backup copies of the installed patches will be saved in
        "/var/adm/patch/backup".

        If you have limited space in /var, you may want to make the backup
        directory the mount point for a separate disk partition, an NFS mounted
        directory, or a symbolic link to another file system.

        You must ensure the backup directory is configured the same way during
        any patch removal operations.

    Your current setup of "/var/adm/patch/backup" is:

     * A plain directory (not a mount point or a symbolic link)
```

   By default, the backup copies of the installed patches will be saved in `/var/adm/patch/backup`. If you have limited space in `/var`, you may want to make the backup directory the mount point for a separate disk partition, an NFS-mounted directory, or a symbolic link to another file system.

5. Answer `yes` when asked if you want to perform the preinstallation check with this setup:

```
    Do you want to proceed with the pre-installation check with this setup? [y]: y
```

6. Enter your name and any information that you want to appear in the preinstallation check log:

```
Your name: Betty

Enter any notes about this operation that you would like stored for
future reference (To end your input, enter a "."):

: preinstall check for patch kit 5
: .
```

7. The program lists any patches that fail the prerequisite and applicability checks, and asks how you want to proceed. You have the following choices:

```
Checking patch prerequisites and patch file applicability...
  (depending upon the number of patches you select, this may take awhile)

   *** The Patch Kit will install 80 patches ***
------------------------------------------------------------------------

Problem installing:

 - Tru64_UNIX_V5.1B / Kernel Patches:
        Patch 26009.00 - SP05 OSFBASE540 (SSRT3631 SSRT3469 SSRT2439 ...)
 ./usr/sbin/cron:
  is installed by Customer Specific Patch (CSP):


 - Tru64_UNIX_V5.1B:
        Patch C 00981.00

  and can not be replaced by this patch. To install this patch,
  ideally, you must first remove the CSP using dupatch.
  Before performing this action, you should contact your
  HP Service Representative to determine if this patch kit
  contains the CSP. If it does not, you may need to obtain a new
  CSP from HP in order to install the patch kit and retain the
  CSP fix.
  or
  you may use dupatch baselining to enable the patch installation.

This patch will not be installed.
------------------------------------------------------------------------

   * The following 1 patch(es) failed in prerequisite/file applicability check:

 - Tru64_UNIX_V5.1B / Kernel Patches:
        Patch 26009.00 - SP05 OSFBASE540 (SSRT3631 SSRT3469 SSRT2439 ...)

   * There were 1 patch(es) which failed in prerequisite/file applicability check:

Press Return to go back to the previous menu
```

2.2 Creating a Baseline

The `dupatch` baselining process looks at the files installed on a system, compares them to the files it expects to find, and prevents the installation of any patch files that might cause an incompatibility among system files. This section provides an overview of the baselining process. See Section 2.2 for instructions on setting a baseline.

Unknown system files occur when the files are replaced through non-standard system file installation methods such as the following:

- The manual installation of system files such as system administration customizations or manually installed patches
- Using the `setld` utility to install system files from user-derived `setld` subsets
- Using the `setld` utility to install files for layered software products
- Changes that result from weak system control programs (usually named *`file`*`.scp`)

Missing system files result from a root user manually deleting system files that were installed during a standard full or update installation procedure or with the dupatch utility. The file is removed but the system inventory records are still in place.

Unknown and missing system files will block patch installations until you take corrective action. However, before taking any action, it is important that you understand the origin of the unknown system files or why missing files are no longer present on your system. Changing the system without this knowledge could leave your operating system or layered product software environment in an inconsistent and nonoperational state.

For example, a file whose origin is unknown that is blocking the installation of a Release patch could be part of a manually installed Customer-Specific Patch (CSP) that is not contained in the Release patch. Removing that one file will disrupt the operation of your CSP and possibly the operation of the system.

When you run the dupatch system baseline feature, a baseline log file is captured in /var/adm/patch/log/baseline.log. (See Appendix A for information about log files.)

You may need to set the patch baseline for your system if you have manually installed system files or if dupatch informs you that patch installation is blocked by system files that are missing or unknown.

Warning

Misusing the baselining feature can cause serious problems with your system. It is important to be aware of the following potential problems:

- Enabling baselining to override its applicability checking could leave your operating system or layered product software environment in an inconsistent and nonoperational state.
- Enabling baselining to update your system sets a new baseline for your operating system or TruCluster software environments. You will not be able to revert to the previous system state for manually installed patches that were marked as installed by baselining.

  We recommend that you backup your /, /usr, and /var file systems before enabling system updates through dupatch baselining.

Baselining is divided into seven phases that provide system information and optionally allow you to take actions that change the patch baseline of your system. You can run through all phases of baselining to get the system analysis without enabling changes to your system. You can run baselining in multiuser mode when you are the root user.

2.2.1 Phase 1 – System Evaluation

The primary goal of Phase 1 is to evaluate your system relative to the patch kit that is being installed. However, the baselining feature will report all missing and unknown files to assist you in better understanding the state of the changed files on the system.

The rest of the baselining phases use the information gathered in Phase 1 to inform you of any installation conflicts for patches contained in the patch kit.

The amount of time needed to evaluate the state of the system varies depending on the size of the patch kit, the version of the software product, and the performance of the system.

2.2.2 Phase 2 – Patch Layered Product Conflicts

Phase 2 reports information for patches whose installation is blocked by system files that were installed by layered products.

Baselining will not override layered product patch installation collision detection mechanisms as it is likely that the layered product or application customizations are not contained in the patch. Installation of the patch in this situation would leave the layered product or application nonoperational.

To resolve this situation, contact your layered product or application Customer Services or HP Services if you have purchased Business Critical Services.

2.2.3 Phase 3 – Identifying Manually Installed Patches

Phase 3 reports patches that exactly match existing files on your system that are not marked as *installed* by the system inventory. For example, in earlier kits, TruCluster software Release patches were installed manually. This phase will report any manually installed Release patch files that exactly match a patch contained in the current `dupatch`-based TruCluster software patch kit.

You can optionally enable `dupatch` to mark these patches as *installed*, which involves copying valid `setld` database information to your system. The `dupatch` utility will copy the appropriate *`patch_subset`*`.inv`, *`patch_subset`*`.scp`, and *`patch_subset`*`.ctrl` files into place for these patches.

If you do not want to enable `dupatch` to mark these patches as installed, you must manually remove the patched system files so the normal `dupatch` installation can install the affected patches.

2.2.4 Phase 4 – Handling Missing or Unknown Files on Your System

Phase 4 reports information about any unknown and missing system files. These files should be considered as intentional customizations which are important to correct system operation. As such, care should be taken to understand why system files have been customized.

Before enabling any patch installations in Phase 5, review the information reported in Phase 4 against your log of manual system changes to ensure you understand why the system was intentionally customized and to determine how to proceed. In some cases you may need to remove customizations to ensure proper system operation.

To assist you in identifying the origin of changed system files, baselining now reports all missing or unknown system files.

The following sections provide general guidance for some of the normal situations where system files are intentionally customized manually.

2.2.4.1 Manually Installed CSPs

In response to a problem report, you may receive a manually installable CSP from your service provider. CSPs are a set of compatible files that deliver fixes to the problems you reported. Additionally, the patch may include instrumentation necessary for debugging purposes.

If your system was customized through a manual installation of CSPs, you must ensure that the fixes delivered by the CSPs are included in the current Release Patch Kit before enabling `dupatch` to overwrite any unknown or missing system files.

Warning

If you are unsure if the CSP is included in the Release Patch Kit, do not enable dupatch to overwrite the manually installed CSP. If you must install the Release patch being blocked by a CSP, contact your service provider for assistance.

If the unknown or missing files are attributable to manually installed CSPs that are included in a Release Patch Kit, perform one of the following steps:

- If all CSP files are overwritten by the patches noted in Phase 5, you can safely enable dupatch to overwrite applicable missing or unknown system files.
- If some of the CSP files are not overwritten by the patches noted in Phase 5, contact your service provider for assistance.

To determine if your CSP is included in the Release Patch Kit, refer to the *Patch Summary and Release Notes* for the Release Patch Kit. See Patch Process Resources and Related Documentation for information about viewing patch documentation on the Web.

2.2.4.2 Manually Installed Release Patches

For some software products, manual installation has been the practiced method for patch installation. For example, patches for TruCluster software used to be installed manually.

You must determine whether the fixes delivered by the manually installed Release patches are included in the current dupatch-based Release Patch Kit before enabling dupatch to overwrite any unknown or missing system files. Once you have made this determination, proceed as follows:

- If the unknown or missing system files are attributable to the manual installation of Release patches and those patches are included in the current dupatch-based Release Patch Kit, you can safely enable dupatch to overwrite applicable missing or unknown system files.
- If the unknown or missing system files are not attributable to manual installation, you must understand the origin of the unknown or missing system files by reviewing the information reported in Phase 4 against your log of manual system changes to ensure you understand why the system was intentionally customized, and to determine how to proceed.

2.2.4.3 User Customized Commands and Utilities

Periodically, system administrators of production computing environments replace Tru64 UNIX commands or utilities with freeware or their own customized version of the command or utility. In this situation you must ensure the unknown or missing files are attributable to intentional replacement of commands, utilities, or other system files.

If the unknown or missing system files are attributable to the replacement of commands, utilities, or other system files with customized versions for the computing environment, do not enable dupatch to overwrite the manually installed customized files. Instead, determine the reason for the customization and then decide how to proceed.

2.2.5 Phase 5 – Enabling dupatch to Overwrite Changed System Files

Phase 5 reports patches that are blocked due to missing or unknown system files, and optionally allows you to override the dupatch conflict management mechanism so the dupatch-based patch may be installed.

For each patch that is blocked by a missing or unknown system file you are presented with the following information:

- Software product identifier
- Patch category
- Patch identifier
- Patch subset description
- The list of unknown and missing files that block the patch installation
- The origin of all other files contained in the patch

Optionally, you can enable `dupatch` to override the collision detection mechanisms and install any of these patches. Use the missing and unknown file information presented in Phase 4 and your system administration log of manual system changes to make Phase 5 patch installation enabling decisions.

We recommended that you do not enable `dupatch` to install patches over missing or unknown system files for which you do not know the origin. Doing so may leave your operating system and TruCluster software environment in an inconsistent and nonoperational state.

We also recommend that you backup your operating system prior to the actual patch installation.

2.2.6 Phase 6 – Report CSPs with Inventory Conflicts

Phase 6 provides the information about patches that have inventory conflict due to certain CSPs that are installed on the system. You will use this information when considering your decision in Phase 7.

2.2.7 Phase 7 – Enable patches with File Applicability Conflicts

Phase 7 allows you to install patches whose inventory does not match the installed system when the system file changed originates from a CSP.

Failing to determine the origin of the files that are in conflict can cause your operating system to be compromised. We therefore recommend that you track down the origin of those files.

To assist you in this effort, this phase lists the additional files that have been installed with the files that cannot be superseded. You can run through this phase to get the analysis without enabling the installation of any of the listed patches.

2.2.8 Steps for Running the Baseline Procedure

The following steps begin the baseline procedure:

1. Log in as root.
2. Run `dupatch` and enter `5` in response to the `Enter your choice` prompt of the Main Menu:

```
Tru64 UNIX Patch Utility (Rev. 48-00)
=========================
        - This dupatch session is logged in /var/adm/patch/log/session.log

    Main Menu:
    ---------

    1)  Patch Kit Installation
    2)  Patch Kit Deletion
    3)  Patch Kit Documentation

    4)  Patch Tracking
    5)  Patch Baseline Analysis/Adjustment
```

```
    h)  Help on Command Line Interface

    q)  Quit

 Enter your choice: 5
```

3. Enter the location of the patch distribution:

```
Enter path to the top of the patch distribution,
or enter "q" to get back to the menu [/patches/PK4/patch_kit]:
```

The baselining procedure then runs through it's seven phases as follows:

- Baselining Phase 1 evaluates your system relative to the patch kit.
- Baselining Phase 2 reports information for patches whose installation is blocked by system files that were installed by layered products. You cannot enable dupatch to install patches that replace system files installed by layered products. You must contact your layered product customer services or HP Services if you have purchased Business Critical Services.
- Baselining Phase 3 reports on patches that match existing files on your system, but are not marked as *installed* by the system inventory. You can tell dupatch to mark these patches as *installed*. This involves copying valid setld database information to your system. If exact matches are found you will be asked the following question:

  ```
  Do you want to mark these patches as installed ? [y/n]
  ```

  You must provide an answer; there is no default answer.
- Baselining Phase 4 reports information about any unknown or missing system files. This information is provided to assist you in understanding the state of files that may prevent patch installation.

  Consider this information carefully when making decisions to override patch-installation checks for patches noted in Phase 5.
- Phase 5 reports patches that do not pass installation applicability tests due to the current state of your system. The installation of these patches is prevented by missing or unknown system files.

  The dupatch utility reports the known information about the files contained in each patch and asks if you want to enable the installation:

  ```
  Do you want to enable the installation of any of these patches? [y/n]:
  ```

  Answer n until you know the origin of the files that are preventing the patch installation. The changed system files that are preventing the Release patch installation may be part of a manually installed Customer-Specific patch or an intentionally customized utility or file.

  If, for example, the file that is preventing the installation of a Release patch is one of many files that are part of a CSP, you must determine how to proceed. For more information, see Section 2.2.4.1 and Section 2.2.5.

  If you answer y to this question, dupatch enables all of the patches to be installed.
- Phase 6 reports CSPs that have inventory conflicts. Some CSPs replace files delivered in the original operating system inventory or other layered products' inventory. The dupatch utility blocks the installation of those patches with inventory conflicts because they can compromise the integrity of the CSPs.

  ```
    Press RETURN to see the list of patches...

     * list of Customer Specific Patches with inventory conflicts:
       ------------------------------------------------------------

  - Tru64_UNIX_V5.1B / Kernel Patches:
        Patch 26009.00 - SP05 OSFBASE540 (SSRT3631 SSRT3469 SSRT2439 ...)
  ```

```
   - Files with Customer Specific Patch conflicts are:

       ./usr/sbin/cron is shipped by:

                 Product: "Tru64 UNIX V5.1B Patch Distribution"
                 (T64KIT0024386-V51BB25-20041206OSF540,06-Dec-2004:04:41:29)
                 Subset:  OSFPATC0098100540


   - Other file(s) within this patch, with their origin (identified
     through checksum match) listed in terms of their translated
     subset information, if any, are:

       ./etc/.new..magic
      Base System

       ./etc/.new..nsswitch.conf
      Tru64_UNIX_V5.1B Patch   26009.00
.
.
.
    Press RETURN to proceed to the next phase...
```

- Phase 7 lists additional files that have been installed with the files in the CSP or layered product that cannot be superseded .

  After reviewing this section, you can enable the installation of these patches. Enabling a patch means that the checks for patch file applicability, done during patch installation, will be bypassed if you choose to install that patch:

```
   It is recommended that you understand the origin of the
   listed files before enabling a patch for installation.

   Press RETURN to see the list of patches...

   OSFPATC0098100540   CONFLICTING FILE   ./usr/sbin/cron


    * Enabling a patch for installation means allowing to modify
      these files. It is recommended that you understand the origin
      of the listed files before enabling a patch for installation.

Do you want to enable the installation of these patches? [y/n]: y

*** Installation of the following patches is enabled:
    (NOTE: You need to include these patches for installation
           from the installation menu)

 - Tru64_UNIX_V5.1B / Kernel Patches:
       Patch 26009.00 - SP05 OSFBASE540 (SSRT3631 SSRT3469 SSRT2439 ...)

* Baseline Analysis/Adjustment process completed.
  ============================================

Press RETURN to get back to the Main Menu...
```

3

Patch Installation and Removal Instructions

This chapter provides instructions for installing and removing patches from the Tru64 UNIX operating system and the TruCluster software products. Although the descriptions and examples in this chapter reflect the installation and removal steps of Release Patch Kits, the steps are basically the same for `dupatch`-based CSP and ERP kits.

Chapter 4 describes the procedure for patching a TruCluster Server Version 5.0A or higher cluster using the rolling upgrade function. If you are patching your system with that process, follow the steps described there. You will be returned to this chapter when it is time to run `dupatch`.

If you have not yet created your cluster, follow the steps in Section 1.8.3.

The `-l` of the `setld` command is disabled for patch subsets.

3.1 Before You Begin the Installation

Before beginning the installation, make sure that you have completed all of the following preliminary steps:

- Make sure you have the correct software

  You must have the appropriate versions of Tru64 UNIX and TruCluster software installed on your system to install patch kits. There are separate patch kits for each version of the Tru64 UNIX and TruCluster software products. The patch kits will not install on any other version of those products. For example, a Tru64 UNIX 5.1B patch kit will only install on Tru64 UNIX Version 5.1B.

- Back up your system

  It is recommended that you backup your `/`, `/usr`, and `/var` file systems prior to installing patches or baselining your system.

- Make sure you have enough storage space

  Refer to the *Patch Summary and Release Notes* for the required storage space.

- Make the patch distribution available to your system, as described in Section 3.2.
- Load any new patch tools, as described in Section 1.2.2.
- Perform the patch preinstallation check, as described in Section 2.1.
- Set a system patch baseline, if needed, as described in Section 2.2.
- Review the list of issues and restrictions in Section 1.8 and in the *Patch Summary and Release Notes* document that comes with your patch kit.

The following sections provide step-by-step instructions for installing and enabling patches.

3.2 Expanding the Patch Kit Tar File

If you are using patch tar files obtained via the Internet (see Patch Process Resources), you must expand the tar file to access the patch kits. The tar file can be expanded on any mountable file system. The following list describes procedure:

1. Mount the file system and create a directory.

```
# /usr/sbin/mount /dev/disk/dsk3g /patches
# cd /patches
# mkdir pk5
```

Note

If you are installing successive patch kits, place and untar each kit in a separate directory.

Copy or ftp the patch kit to the directory you created. For example:

```
# cp T64V51BB26AS0005-20050215.tar  /patches/pk5
```

2. Untar the patch kit, capturing the process to a log file. For example:

```
# script untar.log
# tar -xpvf /patches/pk5/T64V51BB26AS0005-20050215.tar
# Ctrl/d
```

3. View the `untar.log` for errors or failures untarring the file.

3.3 Choosing Single-User or Multiuser Mode

You can install patches from either single-user or multiuser modes. See Section 1.8.1 for information about selecting one of these modes. Section 3.3.1 describes the process from single-user mode and Section 3.3.2 describes the process from multiuser mode. Section 3.4 describes the remaining steps, which are common to installations from single-user and multiuser modes.

3.3.1 Installing Patches from Single-User Mode

The following steps describe a patch kit installation from single-user mode. Although these steps are the same whether installing an old or new style patch kit, the text that `dupatch` displays differs in minor ways. The examples used in these steps reflect the output of a new style patch kit installation.

1. Halt the system. For example:

```
# /usr/sbin/shutdown -h +5 "Applying 5.1B-3 OS and TCR  patches"
```

2. Boot to single-user mode from the console prompt. For example:

```
>>>boot -fl s
```

3. Run the `init s` command to change the run level to a single-user state with only essential kernel services:

```
# /sbin/init s
```

4. Run the `bcheckrc` command to check and mount all the UFS and AdvFS file systems, the `kloadsrv` command to load kernel modules into the kernel, and the `lmf reset` command to copy license details for all enabled products from the License Database to the kernel cache:

```
# /sbin/bcheckrc
# /sbin/kloadsrv
# /usr/sbin/lmf reset
```

5. For systems prior to 5.0A, issue the `update` command and activate your swap partition with the `swapon` command:

```
# /sbin/update
# /sbin/swapon -a
```

6. Enter the `rcinet` command to start network services:

```
# /usr/sbin/rcinet start
```

Informational messages will appear on the screen.

7. Run the dupatch utility. You will be asked to specify the path to the patch_kit file. For example:

```
# cd /var/patch/pk5/patch_kit
# ./dupatch

Enter path to the top of the patch distribution,
or enter "q" to quit : .
```

8. From the Main Menu, enter 1 at the Enter your choice prompt to invoke the patch installation session. For example:

```
Tru64 UNIX Patch Utility (Rev. 48-00)
=========================
        - This dupatch session is logged in /var/adm/patch/log/session.log

    Main Menu:
    ---------

    1)  Patch Kit Installation
    2)  Patch Kit Deletion
    3)  Patch Kit Documentation

    4)  Patch Tracking
    5)  Patch Baseline Analysis/Adjustment

    h)  Help on Command Line Interface

    q)  Quit

Enter your choice: 1
```

9. When the patch installation menu is displayed, enter 2 at the Enter your choice prompt:

```
     Patch Installation Menu:
     ------------------------

    1)  Pre-Installation Check ONLY
    2)  Check & Install the patch kit in Single-User Mode

    b)  Back to Main Menu
    q)  Quit

Enter your choice: 2
```

3.3.2 Installing Patches from Multiuser Mode

The following list describes the steps you take and the type of output you will see when you install patches from multiuser mode.

1. Run the dupatch utility and enter 1 at the Enter your choice prompt to the invoke the patch installation session:

```
# /patches/pk4/patch_kit/dupatch
Tru64 UNIX Patch Utility (Rev. 48-00)
=========================
        - This dupatch session is logged in /var/adm/patch/log/session.log

    Main Menu:
    ---------

    1)  Patch Kit Installation
    2)  Patch Kit Deletion
    3)  Patch Kit Documentation

    4)  Patch Tracking
    5)  Patch Baseline Analysis/Adjustment

    h)  Help on Command Line Interface

    q)  Quit

     Enter your choice: 1
```

2. When the patch installation menu is displayed. Enter 3, at the `Enter your choice` prompt. Read the warning message and press Return if you want to continue the installation in multi-user mode:

```
     Patch Kit Installation Menu:
     ----------------------

    1)  Pre-Installation Check ONLY
    2)  Check & Install in single-user mode w/ network services
    3)  Check & Install in Multi-User mode

    b)  Back to Main Menu
    q)  Quit

Enter your choice: 3

                *** Installation Warning ***

You have chosen to install the patch kit onto this system while it is
running in Multi-User mode. Some patches may directly affect core operating
system operations. To ensure the proper operation of all applications, it is
strongly suggested that you install these patches while the system is in
Single-User mode. If this cannot be done, install these patches when the
system is as lightly loaded as possible (i.e. not running production
environments, no users logged on, etc.).

Do you wish to continue? (y/n) [y]:
```

3.4 Common Installation Steps

The following steps provide instructions for continuing the installation of Tru64 UNIX and TruCluster software patches after you have selected either single-user or multiuser mode.

1. Specify whether or not you accept the license agreement. You can read the license on screen or you can read the license before beginning the installation process in the *Patch Summary and Release Notes* that comes with the patch kit. In the following output, the license is removed to save space:

```
Tru64 Unix License Agreement
** ... **

To read the license again, type 'license'.
Do you accept the license agreement? (y/n) : y

Checking patch kit for transmission errors during download...

Finished Checking patch kit checksums
```

2. You have the option to make patches reversible so you can return the system to its state prior to the installation of a patch. Enter `y` or press Return to make the patches reversible. For example:

```
Do you want the patches to be reversible? [y]:
```

   By default, backup copies of the installed patches are saved in `/var/adm/patch/backup`. If you have limited space in `/var`, you may want to make the backup directory the mount point for a separate disk partition, an NFS-mounted directory, or a symbolic link to another file system.

   If you answer no to this question, the existing system files will not be saved and the installed patches will not be reversible. HP recommends that you install patches so they are reversible.

3. The program describes your backup setup and asks you if you want to proceed:

```
Do you want to proceed with the installation with this setup? [y]:
```

4. You are asked to record your name as the person installing the patches and to add any comments you would like stored for future reference. For example:

```
Your name: Joe C.
```

   Enter any notes about this operation that you would like stored for future reference. To end your input, enter a period (.) and press Return.

```
: Installing Patch Kit 5
: . Return
```

5. The next action depends on the type of kit you are installing:

   - Inclusive patch kit

     With this type of kit dupatch peforms a preinstallation check and begins to install the patches if it finds no problems. For example:

     ```
     Checking patch prerequisites and patch file applicability...
       (depending upon the number of patches you select, this may
        take awhile).

        *** Installing 78 patches ***
     ```

     If any patches fail the preinstallation check, do one of the following:

     - If the failure is the result of a file conflict, you will need to run the patch baseline process, as described in Section 2.2.
     - If the failure is caused by an installed CSP that is not included in the current patch kit, you will have to remove the CSP, install the patch kit, and reinstall the CSP. See Section B.1.8 for more information.

   - Customer-Specific patch kit

     With this type of kit you must install all patches. You can, however, remove individual CSPs after the installation process is completed and the system has been rebooted.

3.5 Rebuilding the Kernel

The dupatch utility determines whether the installation or removal of patches requires that the kernel be rebuilt. This action is performed automatically or manually, depending upon the method you used to install the patches:

- When using the menu-based interface, you will be prompted for actions to take. Those prompts are the same ones you would see if you ran the doconfig command. The dupatch utility asks if your system has a custom configuration file and if you want to change it.
- When using dupatch from the command line, the kernel is built automatically. It does this by calling the doconfig -a command. If you specify the dupatch -cfgfile command, dupatch calls doconfig with the -a-c options.

After the patch kit is installed you will see output similar to the following:

```
Configuring "Patch: SP04 OSFADVFSBIN540" (OSFPAT02500300540)

Configuring "Patch: SP04 OSFADVFS540 (SSRT2275)" (OSFPAT02500200540)

Beginning kernel build...


Do you have a pre-existing configuration file?:
```

If you answer yes, dupatch will build the kernel noninteractively, enabling all (mandatory and optional) kernel options automatically. This procedure is similar to running the doconfig -a command.

If you answer no, dupatch will build the kernel interactively. This procedure is similar to running the doconfig -c command. The following steps describe this procedure and provide some guidance for making your selections:

1. Enter a new name for the kernel configuration file or accept the default. If you accept the default you will be asked if you want to replace it. For example:

   ```
     *** KERNEL CONFIGURATION AND BUILD PROCEDURE ***

   Enter a name for the kernel configuration file. [IDIOM2]: Return
   ```

```
A configuration file with the name 'IDIOM2' already exists.
Do you want to replace it? (y/n) [n]: y
Saving /sys/conf/IDIOM2 as /sys/conf/IDIOM2.bck
```

2. Specify the kernel options you want. If you are unsure of which options to specify, consider the following:

   - Selecting the `All of the Above` option ensures that you can access any new functions provided by the patch kit. You may, however, create a kernel that is larger than you need.

     If you know of options you do not need, you can ignore those and specify all of the other options, thereby ensuring that you will have access to the new functions you need but with a smaller kernel than if you had selected all of the options.

   - Selecting the `None of the Above` option will result in a kernel build that is similar to using the `doconfig -ac` command. This is the default.

   The following output is similar to what you will see. The procedure gives you the opportunity to edit the configuration file:

```
*** KERNEL OPTION SELECTION ***

    Selection   Kernel Option
-----------------------------------------------------------------
 1 System V Devices
 2 NTP V3 Kernel Phase Lock Loop (NTP_TIME)
 3 Kernel Breakpoint Debugger (KDEBUG)
 4 Packetfilter driver (PACKETFILTER)
 5 IP-in-IP Tunneling (IPTUNNEL)
 6 IP Version 6 (IPV6)
 7 Point-to-Point Protocol (PPP)
 8 STREAMS pckt module (PCKT)
 9 Data Link Bridge (DLPI V2.0 Service Class 1)
 10 X/Open Transport Interface (XTISO, TIMOD, TIRDWR)
 11 Digital Versatile Disk File System (DVDFS)
 12 ISO 9660 Compact Disc File System (CDFS)
 13 Audit Subsystem
 14 ATM UNI 3.0/3.1 ILMI (ATMILMI3X)
--- MORE TO FOLLOW ---
Enter your choices or press <Return>
to display the next screen.

Choices (for example, 1 2 4-6): 2-12
 15 IP Switching over ATM (ATMIFMP)
 16 LAN Emulation over ATM (LANE)
 17 Classical IP over ATM (ATMIP)
 18 ATM UNI 3.0/3.1 Signalling for SVCs (UNI3X)
 19 Asynchronous Transfer Mode (ATM)

The following choices override your
previous selections:

 20 All of the above
 21 None of the above
 22 Help
 23 Display all options again
-----------------------------------------------------------------


Enter your choices, choose an overriding action or
press <Return> to confirm previous selections.

Choices (for example, 1 2 4-6): Return

You selected the following kernel options:
 NTP V3 Kernel Phase Lock Loop (NTP_TIME)
 Kernel Breakpoint Debugger (KDEBUG)
 Packetfilter driver (PACKETFILTER)
 IP-in-IP Tunneling (IPTUNNEL)
 IP Version 6 (IPV6)
 Point-to-Point Protocol (PPP)
 STREAMS pckt module (PCKT)
 Data Link Bridge (DLPI V2.0 Service Class 1)
 X/Open Transport Interface (XTISO, TIMOD, TIRDWR)
 Digital Versatile Disk File System (DVDFS)
 ISO 9660 Compact Disc File System (CDFS)
```

```
Is that correct? (y/n) [y]: Return

Do you want to edit the configuration file? (y/n) [n]: Return


*** PERFORMING KERNEL BUILD ***

A log file listing special device files is located in /dev/MAKEDEV.log
 Working....Tue Mar  9 11:36:33 EST 2004

The new kernel is /sys/IDIOM2/vmunix
```

See the doconfig(8) reference page for more information.

3.6 Rebooting the System

The action that dupatch takes to reboot your system depends upon whether you used the command-line or menu-based interface or performed the action in single-user or multiuser mode. The following sections describe these actions.

Before rebooting, review the dupatch session log, /var/adm/patch/log/session.log, to ensure that the installation was successful. Note any special patch instructions, informational messages, and error messages. Certain patches may require you to take a particular action, such as running a script, before rebooting. (See Appendix A for information about dupatch logs.)

3.6.1 In Single-User Mode

When performing a patch installation or removal in single-user mode from the command line, the system automatically reboots after the command line operation is completed.

When performing a patch installation or removal in single-user mode using the menu-based interface, dupatch asks if you want to reboot the system after the patch installation or removal is completed:

- If you answer yes, the system reboots immediately.
- If you answer no, dupatch returns to the appropriate menu — either installation or removal, depending on the operation.

3.6.2 In Multiuser Mode

When installing patches in multiuser mode from the command line, you are given a message informing you that a reboot is necessary to complete the patch installation. However, the system does not reboot itself.

When installing patches in multiuser mode using the menu-based interface, dupatch gives you three options if a reboot is necessary:

- Reboot now
- Schedule a reboot for a later time
- Do not reboot

3.7 Post-Installation Actions

The following sections describe actions for you to take after you have completed the dupatch installation procedure.

3.7.1 Enabling the Version Switch After Installing a New Style Patch Kit

Some patches may require you to run the versw -switch command to enable the new functions delivered in those patches. (See Section 1.7 for information

about version switches.) You perform this action after dupatch has completed the installation:

```
# versw -switch
```

The new functionality will not be available until after you reboot your system. You do not have to run the versw -switch command, but if you do not, your system will not be able to access the functionality provided in the version switch patches.

3.7.2 Remove Temporary Directory

Once your patch kit is installed, delete the temporary directory in which you expanded the patch kit tar file. For example:

```
# rm -r /Patches/PK4
```

Removing the temporary directory will preclude the possibility of using that directory for subsequent patch kit installations. When performing a patch kit installation, using a directory that contains files from a previous patch kit installation can leave your system in an unstable condition.

Remember that if you want to save the patch kit tar file, remove it from the temporary directory before deleting the directory.

3.7.3 Adding the Worldwide Language Support

Inclusive patch kits provide patches to the Tru64 UNIX Worldwide Language Support subset (WLS). If the WLS subset is installed on your system, the WLS patches will be installed automatically when you install the patch kit. However, if you install the WLS subset after patching your system, you will have to rerun dupatch to install the WLS patches. The dupatch utility will see the WLS subset, recognize that the patches have not been installed, and will install them.

3.8 Removing Patches

To remove patches from your system, use the Patch Deletion option of the dupatch Main Menu. The following sections describe actions describe the patch removal process.

3.8.1 Overview

Beginning with the version of dupatch delivered in the Version 5.1B-3 kit, the patch removal process depends upon whether you installed the new form of patch kits, called Inclusive Patch Kits. These kits began shipping with Version 5.1B-2.

With Inclusive Patch Kits you must remove the entire kit rather than individual patches. However, once you have removed any Inclusive Patch Kits installed on your system, you can then remove individual patches from earlier kits.

To do this, dupatch recognizes the type of kit you have installed. When you select the patch deletion menu, dupatch lists the most current Inclusive Patch Kit installed on your system as well as any customer-specific patches (CSPs) that depend upon that kit.

After you remove that kit and reboot your system, you can rerun dupatch to remove the next most current Inclusive Patch Kit and the CSPs that depend on it.

Once all inclusive patch kits have been removed, the next time you run the patch deletion program, dupatch will list all of the patches on your system and you can selectively remove any of those patches.

Caution

With the old-style patch kits, the Patch Deletion menu lists every `setld`-based patch on your system, regardless of which patch kit installed them. If you select the `ALL of the above` menu item, it will remove all `setld`-based patches from your system. Therefore, you want to remove all of the patches from a patch kit, but do not want to delete `setld`-based patches, you will have to specify the patch ID of all of that kit's patches.

The latest version of `dupatch` also gives you the option to delete patches in single-user mode or in multiuser mode. As with the installation process, using single-user mode is safer and is the recommended procedure. See Section 1.8.1 for more information.

The `dupatch` utility issues the following warning when you are deleting patches in multiuser mode.

```
*** Multi-User Deletion Warning ***

You have chosen to delete patches from this system while it is running in
Multi-User mode. Some patches may directly affect core operating system
operations. To ensure the proper operation of all applications, it is
strongly suggested that you delete these patches while the system is in
Single-User mode. If this cannot be done, delete these patches when the
system is as lightly loaded as possible (i.e. not running production
environments, no users logged on, etc.).

Do you want to continue? (y/n):
```

If you want to continue, answer yes. If you do not want to delete the patch kit in multiuser mode, answer no and bring your system down to single-user mode as described in Section 3.3.1.

3.8.2 Important Tasks Required Before Removing Patches and Rebooting System

Before running the patch deletion process you may have to perform the tasks described in the following sections.

3.8.2.1 Run Mandatory Script Before Removing New Style Patch Kits

If you enabled version switches as described in Section 3.7.1 for an Inclusive Patch Kit, you must run the `/usr/sbin/versw_enable_delete` script before attempting to remove the patch kit. The steps for running this script require a complete cluster or single system shutdown, so choose a time when a shutdown will have the least impact on your operations. The following steps describe the procedure:

1. Make sure that all phases of the patch kit installation process have been completed.
2. Run `/usr/sbin/versw_enable_delete`:

   ```
   # /usr/sbin/versw_enable_delete
   ```

3. Shut down the entire cluster or the single system.
4. Reboot the entire cluster or the single system.
5. Run `dupatch` on your single system or on a cluster using the rolling upgrade procedure to delete the patch kit.

Note

The next step requires that you reboot each cluster member to remove the patch kit. Because the no-roll procedure automatically reboots the system after deleting the patches, you would not be able to perform the next steps as required. Therefore, you cannot use the no-roll procedure to remove this patch kit.

6. Reboot the single system or each member of the cluster.

3.8.2.2 Changes to System May Need to Be Reversed

If you made the following changes to your system after installing the patch kit, you will have to undo those changes before you can uninstall the patch kit:

- If you changed your hardware configuration (for example, by adding a new disk), the system configuration that existed prior to installing the patch kit might not recognize the new devices or may not provide the necessary support for them.
- If you added new cluster members, the new members will not have an older state to revert to if you attempt to uninstall the patch kit.

To uninstall the patch kit, do the following:

1. Remove all new hardware and new cluster members that you added after installing the patch kit.
2. Run `dupatch` to uninstall the patch kit.
3. Verify that the patch kit was successfully uninstalled.

You can now add the cluster members you removed and reinstall the hardware you removed, as long as the support for it existed in the pre-patched system. You can also reinstall the patch kit.

3.8.2.3 Script Must Be Run Prior to Reboot on Certain Version 5.1B Systems

If removing a PK4 or higher patch kit restores your Version 5.1B system to a pre-patched state, you must run the script `/etc/dn_fix_dat.sh` before rebooting your system during the patch deletion process. This would occur if the inclusive patch kit you are uninstalling is the only patch kit installed on your Version 5.1B system

You must also run this script if you are removing a specific patch from previous Version 5.1B patch kits if those kits are the only patch kit on your system. The affected patch in those kits will be noted in a Special Instruction that is displayed when you run the `dupatch` installation and deletion processes.

Failing to run this script will result in your system being unable to boot normally. If this occurs, do the following:

1. Boot your system in single-user mode:

```
>>> boot -fl s
```

2. Run the script:

```
# /etc/dn_fix_dat.sh
```

3. Reboot normally.

If you also need to reverse the version switch as described in Section 3.8.2.1, run the `/etc/dn_fix_dat.sh` script after step 5 in that process.

3.8.3 Running dupatch to Remove Patches

The process for removing patches is similar to the one for installing them.

The following steps describe the patch removal process for an Inclusive Patch Kit with the system running in single-user mode. In mutiuser mode the steps would be the same except you would see the multiuser deletion warning described in Section 3.8.1.

See Section 3.3.1 for the steps on bringing down your system to single-user mode.

1. Run dupatch and select 2 for patch removal:

```
# /patch/pk4/patch_kit/dupatch

Tru64 UNIX Patch Utility (Rev. 48-00)
=========================
        - This dupatch session is logged in /var/adm/patch/log/session.log

    Main Menu:
    ---------

    1)  Patch Kit Installation
    2)  Patch Kit Deletion
    3)  Patch Kit Documentation

    4)  Patch Tracking
    5)  Patch Baseline Analysis/Adjustment

    h)  Help on Command Line Interface

    q)  Quit

Enter your choice: 2
```

2. Select the current patch kit. This menu will change if no Inclusive patch kits are installed.

```
There may be more patches than can be presented on a single
     screen. If this is the case, you can choose patches screen by screen
     or all at once on the last screen. All of the choices you make will
     be collected for your confirmation before any patches are deleted.

     1) CSP C688.00  drag-and-drop or cut-and-paste may fail
     2) CSP C718.00  Debug version of ping
     3) CSP C752.00  page on o/h list panic
     4) CSP C882.00  Fix for memory leak and slowdown in rpc.lockd
     5) T64V51BB26AS0005-20050215 and all CSP's dependent upon it

Or you may choose one of the following options:

     2) ALL of the above
     3) CANCEL selections and redisplay menus
     4) EXIT without deleting any patches

Enter your choices or press RETURN to redisplay menus.

Choices (for example, 1 2 4-6): 7

You are deleting the following patches:

        T64V51BB26AS0005-20050215 and all CSP's dependent upon it

Is this correct? (y/n):y

               ***  Start of Special Instructions  ***


If you delete this patch kit, you MUST run the following script prior to
rebooting your system:  /etc/dn_fix_dat.sh
```

⋮

3. You are asked to record your name as the person removing the patches and to add any comments you would like stored for future reference in the log file. For example:

```
Your name: Betty
```

Enter any notes about this operation that you would like stored for future reference. To end your input, enter a period (.) and press Return.

```
: Uninstalling V5.1B-3
: . Return

Checking patch dependency...
  (depending upon the number of patches you select, this may take awhile)


   *** The Patch Kit will delete 67 patches ***

************************* CAUTION ***********************************

 Interruption of this phase of the operation will corrupt your
 operating system software and compromise the patch database
 integrity.

 DO NOT Ctrl/C, power off your system, or in any other way
 interrupt the patch operation. The patch operation is complete
 when you are returned to the Patch Utility menus.

*********************************************************************
Deleting "Patch: SP05 OSFEXER540" (OSFPAT02603100540).
Deleting "Patch: SP05 OSFEXAMPLES540" (OSFPAT02603000540).
Deleting "Patch: SP05 OSFENVMON540" (OSFPAT02602800540).
.
.
.
```

4. Rebuild the kernel. This step is the same as for the installation process. See Section 3.5 for details.
5. Review the session log to ensure the removal was successful. Note any special patch instructions, informational messages, and error messages. This is especially important to identify any actions that you may have to take (such as running a script) before rebooting your system.
6. Run the script described in Section 3.8.2.3.
7. Reboot the system. See Section 3.6 for details.

4

Rolling Upgrade

A rolling upgrade is a software upgrade of a cluster that is performed while the cluster is in operation. Patching your system is one type of upgrade that can be performed using this procedure. The term “Rolling Patch” is sometimes used to describe the patching process using the Rolling Upgrade procedure. In general, the terms Rolling Patch and Rolling Upgrade are synonymous in this chapter.

In a Rolling Upgrade, one member at a time is upgraded and returned to operation while the cluster transparently maintains a mixed-version environment for the base operating system, cluster, and Worldwide Language Support (WLS) software. Clients accessing services are not aware that a rolling upgrade is in progress.

A rolling upgrade consists of an ordered series of steps, called stages. The commands that control a rolling upgrade enforce this order.

When performing a rolling upgrade, the same procedure is used for patching your system as for upgrading to a new operating system or TruCluster version. The principal difference is that for a rolling patch you use the `dupatch` utility and for a rolling upgrade you use the `installupdate` utility during the install stage.

This chapter provides the same information as the Rolling Upgrade chapter of the *Cluster Installation* manual. It is provided here as a convenience so you can review your patching options in one manual.

Note

If you have not yet created your cluster, we recommend that you patch your system first. See Section 1.8.3 for this time-saving procedure.

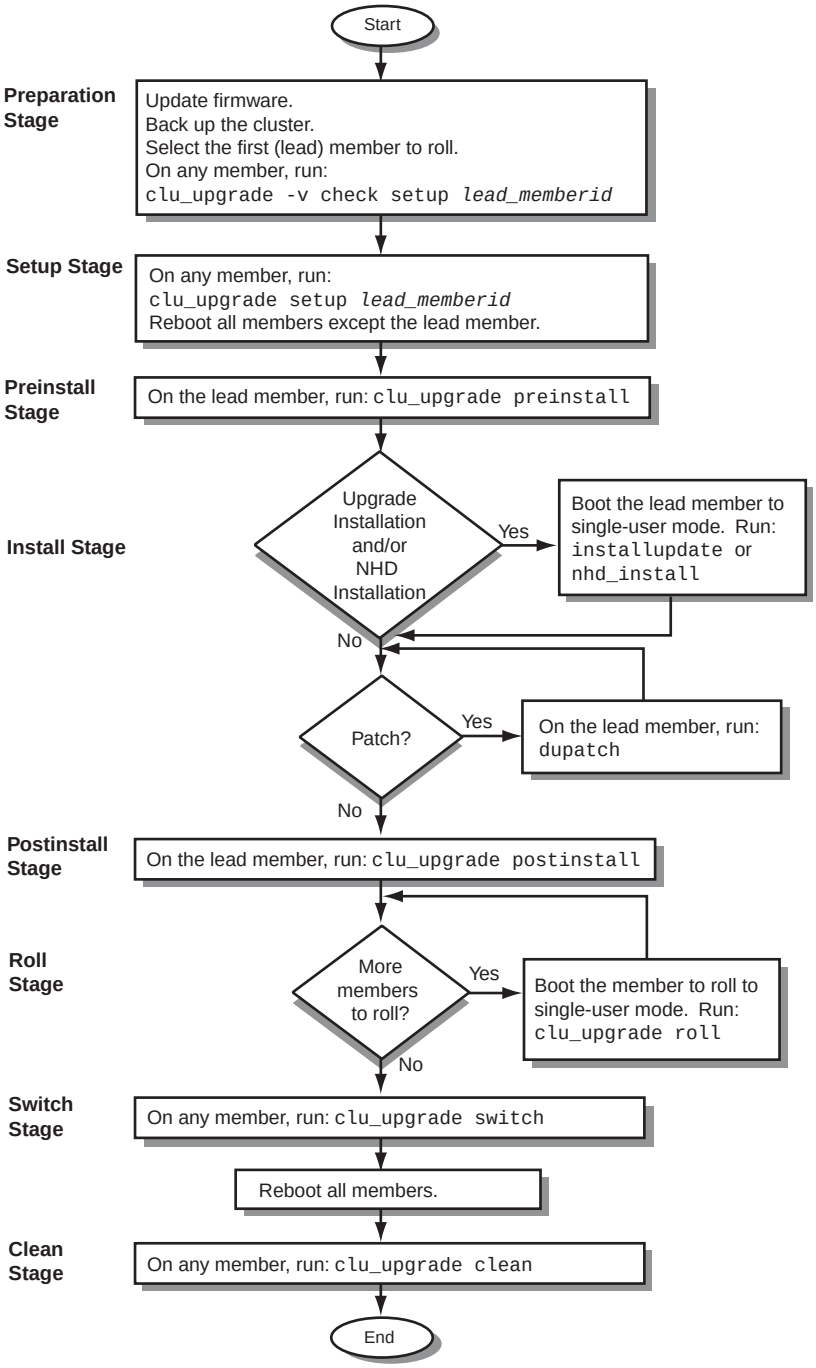
This first part of this chapter contains instructions for performing a rolling upgrade, for displaying the status of a rolling upgrade, and for undoing one or more stages of a rolling upgrade. Those interested in how a rolling upgrade works can find the details in Section 4.7 and the sections that follow it.

This chapter discusses the following topics:

- Tasks, and combinations of tasks, you can perform during a single rolling upgrade (Section 4.1)
- Tasks you cannot perform during a rolling upgrade (Section 4.2)
- How to perform a rolling upgrade (Section 4.3)
- How to display the status of a rolling upgrade (Section 4.5)
- How to undo the stages of a rolling upgrade (Section 4.6)
- The commands used during a rolling upgrade (Section 4.7)
- Rolling upgrade stages (Section 4.8)
- Two mechanisms that support rolling upgrades: tagged files (Section 4.9) and version switches (Section 4.10)
- Rolling upgrade and layered products (Section 4.11)
- Rolling upgrade and RIS (Section 4.12)

Figure 4-1 provides a simplified flow chart of the tasks and stages that are part of a rolling upgrade initiated on a Version 5.1B cluster:

Figure 4-1: Rolling Upgrade Flow Chart



4.1 Rolling Upgrade Supported Tasks

The tasks that you can perform during a rolling upgrade depend on which versions of the base operating system and cluster software are currently running on the cluster. The main focus of this chapter is to describe the behavior of a rolling upgrade that starts on a TruCluster software Version 5.1B cluster. However, because you may read this chapter in preparation for a rolling upgrade from TruCluster software Version 5.1A to Version 5.1B, we point out rolling upgrade differences between the two versions.

The following list describes the basic tasks you can perform within a rolling upgrade:

- Upgrade the cluster's Tru64 UNIX base operating system and TruCluster software software. You perform this type of rolling upgrade to upgrade from the installed version to the next version.

  When performing a rolling upgrade of the base operating system and cluster software, you can roll only from one version to the next version. You cannot skip versions.

  Note

  A rolling upgrade updates the file systems and disks that the cluster currently uses. The roll does not update the disk or disks that contain the Tru64 UNIX operating system used to create the cluster (the operating system on which you ran `clu_create`). Although you can boot the original operating system in an emergency when the cluster is down, remember that the differences between the current cluster and the original operating system increase with each cluster update.

- Patch the cluster's current versions of the Tru64 UNIX base operating system and TruCluster software software.
- Install a New Hardware Delivery (NHD) kit (the cluster must be running TruCluster software Version 5.1A or later).

Rolling in a patch kit or an NHD kit uses the same procedure as rolling in a new release of the base operating system and cluster software. The difference is which commands you run during the install stage:

- To upgrade the base operating system and cluster software, run `installupdate` in the install stage.
- To roll in a patch kit, run `dupatch` in the install stage. You can invoke `dupatch` multiple times in the install stage to roll in multiple patch kits.

  If you want to perform a no-roll patch of the cluster, do not run the `clu_upgrade` command. Instead run the `dupatch` command from a cluster member running in multiuser mode.

  No-roll patching applies patches quickly and reduces the number of reboots required. It patches the cluster in one operation. However, it requires a reboot of the whole cluster to complete the operation, so the cluster is unavailable for a period.
- To install an NHD kit, run `nhd_install` in the install stage.

Throughout this chapter, the term rolling upgrade refers to the overall procedure used to roll one or more software kits into a cluster.

As shown in Figure 4-1, you can perform more than one task during a rolling upgrade.

If the cluster is running Version 5.1A or Version 5.1B, a rolling upgrade can include the task combinations listed in Table 4-1:

Table 4-1: Rolling Upgrade Tasks Supported by Version 5.1A and Version 5.1B

An update installation from Version 5.1A to Version 5.1B	
An update installation from Version 5.1B to the next release	
A patch of Version 5.1A	
A patch of Version 5.1B	

Table 4-1: Rolling Upgrade Tasks Supported by Version 5.1A and Version 5.1B (cont.)

The installation of a New Hardware Delivery (NHD) kit onto a Version 5.1A cluster The installation of an NHD kit onto a Version 5.1B cluster
An update installation from Version 5.1A to Version 5.1B of the base operating system and cluster software, followed by a patch of Version 5.1B An update installation from Version 5.1B to the next release of the base operating system and cluster software followed by a patch of the next release[a]
An NHD installation onto a Version 5.1A cluster followed by a patch of Version 5.1A An NHD installation onto a Version 5.1B cluster followed by a patch of Version 5.1B
An update installation from Version 5.1A to Version 5.1B followed by the installation of an NHD kit for Version 5.1B An update installation from Version 5.1B to the next release of the base operating system and cluster software followed by the installation of an NHD kit for that next release[b]
An update installation from Version 5.1A to Version 5.1B, followed by the installation of an NHD kit for Version 5.1B, followed by a patch of Version 5.1B An update installation from Version 5.1B to the next release, followed by the installation of an NHD kit for the next release, followed by a patch of the next release[b]

[a] Within one rolling upgrade, you can combine an upgrade of the base operating system and cluster software with a patch of the new software. This means that during the install stage, you can run `installupdate` on the first member followed by `dupatch` to patch the newly installed software. When you roll the remaining members they automatically get both the new software and the patches.

However, you cannot patch the current software and then upgrade the base operating system and cluster software within one rolling upgrade. This operation requires two rolling upgrades.

[b] Allowed only if you have already installed an NHD kit on the Version 5.1A or Version 5.1B cluster.

4.2 Unsupported Tasks

The following list describes tasks that you cannot perform or that we recommend you do not attempt during a rolling upgrade:

- Do not remove or modify files in the `/var/adm/update` directory. The files in this directory are critical to the roll. Removing them can cause a rolling upgrade to fail.
- During the install stage, you cannot run a `dupatch` command followed by an `installupdate` command. To patch the current software before you perform a rolling upgrade, you must perform two complete rolling upgrade operations: one to patch the current software, and one to perform the update installation.
- You cannot bypass versions when performing a rolling upgrade of the base operating system and cluster software. You can only roll from one version to the next version.
- Do not use the `/usr/sbin/setld` command to add or delete any of the following subsets:
  - Base Operating System subsets (those with the prefix `OSF`).
  - TruCluster Server subsets (those with the prefix `TCR`).
  - Worldwide Language Support (WLS) subsets (those with the prefix `IOS`).

  Adding or deleting these subsets during a roll creates inconsistencies in the tagged files.
- Do not install a layered product during the roll.

  Unless a layered product's documentation specifically states that you can install a newer version of the product on the first rolled member, and that the layered product knows what actions to take in a mixed-version cluster, we strongly recommend that you do not install either a new layered product or a new version of a currently installed layered product during a rolling upgrade.

For more information about layered products and rolling upgrades, see Section 4.11.

4.3 Rolling Upgrade Procedure

In the procedure in this section, unless otherwise stated, run commands in multiuser mode. Each step that corresponds to a stage refers to the section that describes that stage in detail. We recommend that you read the detailed description of stages in Section 4.8 before performing the rolling upgrade procedure.

Some stages of a rolling upgrade take longer to complete than others. Table 4-2 lists the approximate time it takes to complete each stage.

Table 4-2: Time Estimates for Rolling Upgrade Stages

Stage	Duration
Preparation	Not under program control.
Setup	45 - 120 minutes.[a]
Preinstall	15 - 30 minutes.[a]
Install	The same amount of time it takes to run `installupdate`, `dupatch`, `nhd_install`, or a supported combination of these commands on a single system.
Postinstall	Less than 1 minute.
Roll (per member)	Patch: less than 5 minutes. Update installation: about the same amount of time it takes to add a member.[b]
Switch	Less than 1 minute.
Clean	30 - 90 minutes.[a]

[a] These stages create, verify, or remove the tagged files required for a rolling upgrade. The time that it takes to run one of these stages depends on the speed of the member executing the command, the speed of the storage, and whether the member executing the command is the CFS server for the root (`/`), `/usr`, and `/var` file systems. Consider relocating these file systems to the member where you will run the `clu_upgrade` command.
[b] After rolling the lead member, use parallel rolls to roll multiple members simultaneously and shorten the time it takes to roll a cluster.

You can use the following procedure to upgrade a TruCluster software Version 5.1A cluster to Version 5.1B, and to upgrade a cluster that is already at Version 5.1B.

1. Prepare the cluster for the rolling upgrade (Section 4.8.1):

   a. Choose one cluster member to be the lead member (the first member to roll). (The examples in this procedure use a member whose `memberid` is `2` as the lead member. The example member's host name is `provolone`.)

   b. Back up the cluster.

   c. If you will perform an update installation during the install stage, remove any blocking layered products, listed in Table 4-6, that are installed on the cluster.

   d. To determine whether the cluster is ready for an upgrade, run the `clu_upgrade -v check setup` *`lead_memberid`* command on any cluster member. For example:

      ```
      # clu_upgrade -v check setup 2
      ```

      If a file system needs more free space, use AdvFS utilities such as `addvol` to add volumes to domains as needed. For disk space requirements, see Section 4.8.1. For information on managing AdvFS domains, see the Tru64 UNIX *AdvFS Administration* manual.

e. Verify that each system's firmware will support the new software. Update firmware as needed before starting the rolling upgrade.

2. Perform the setup stage (Section 4.8.2).

Notes

If your current cluster is at Version 5.1A or later and if you plan to upgrade the base operating system and cluster software during the install stage, mount the device or directory that contains the new TruCluster software kit before running `clu_upgrade setup`. The `setup` command will copy the kit to the `/var/adm/update/TruClusterKit` directory.

If your current cluster is at Version 5.1A or later and if you plan to install an NHD kit during the install stage, mount the device or directory that contains the new NHD kit before running `clu_upgrade setup`. The `setup` command will copy the kit to the `/var/adm/update/NHDKit` directory.

On any member, run the `clu_upgrade setup` *`lead_memberid`* command. For example:

```
# clu_upgrade setup 2
```

Section 4.8.2 shows the menu displayed by the `clu_upgrade` command.

When the setup stage is completed, `clu_upgrade` prompts you to reboot all cluster members except the lead member.

3. One at a time, reboot all cluster members except the lead member. Do not start the preinstall stage until these members are either rebooted or halted.

4. Perform the preinstall stage (Section 4.8.3).

On the lead member, run the following command:

```
# clu_upgrade preinstall
```

If your current cluster is at Version 5.1A or later, the `preinstall` command gives you the option of verifying or not verifying the existence of the tagged files created during the setup stage.

- If you have just completed the setup stage and have done nothing to cause the deletion any of the tagged files, you can skip this test.
- If you completed the setup stage a while ago and are not sure what to do, let `preinstall` test the correctness of the tagged files.

5. Perform the install stage (Section 4.8.4).

Note

During the install stage you load the new software on the lead member, in effect rolling that member. When you perform the roll stage, this new software is propagated to the remaining members of the cluster.

The `clu_upgrade` command does not load software during the install stage. The loading of software is controlled by the commands you run: `installupdate`, `dupatch`, or `nhd_install`.

See Table 4-1 for the list of rolling upgrade tasks and combination of tasks supported for Version 5.1A and Version 5.1B.

a. See Chapter 3 for instructions on installing a patch kit using the `dupatch` command.

   See the Tru64 UNIX *Installation Guide* for detailed information on using the `installupdate` command.

   See the Tru64 UNIX *New Hardware Delivery Release Notes and Installation Instructions* that came with your NHD kit for detailed information on using the `nhd_install` command.

b. If the software you are installing requires that its installation command be run from single-user mode, halt the system and boot the system to single-user mode:

```
# shutdown -h now
>>> boot -fl s
```

Note

Halting and booting the system ensures that it provides the minimal set of services to the cluster and that the running cluster has a minimal reliance on the member running in single-user mode. In particular, halting the member satisfies services that require the cluster member to have a status of DOWN before completing a service failover. If you do not first halt the cluster member, services will probably not fail over as expected.

   When the system reaches single-user mode, run the following commands:

```
# init s
# bcheckrc
# lmf reset
```

c. Run the `installupdate`, `dupatch`, or `nhd_install` command.

   To roll in multiple patch kits, you can invoke `dupatch` multiple times in a single install stage. Be aware that doing so may make it difficult to isolate problems should any arise after the patch process is completed and the cluster is in use.

   You cannot run a `dupatch` command followed by an `installupdate` command. To patch the current software before you perform a rolling upgrade, you must perform two complete rolling upgrade operations: one to patch the current software, and one to perform the update installation.

6. (Optional) After the lead member performs its final reboot with its new custom kernel, you can perform the following manual tests before you roll any additional members:

   a. Verify that the newly rolled lead member can serve the shared root (`/`) file system.

      i. Use the `cfsmgr` command to determine which cluster member is currently serving the root file system. For example:

```
# cfsmgr -v -a server /

 Domain or filesystem name = /
 Server Name = polishham
 Server Status : OK
```

ii. Relocate the root (/) file system to the lead member. For example:

```
# cfsmgr -h polishham -r -a SERVER=provolone /
```

b. Verify that the lead member can serve applications to clients. Make sure that the lead member can serve all important applications that the cluster makes available to its clients.

You decide how and what to test. We suggest that you thoroughly exercise critical applications and satisfy yourself that the lead member can serve these applications to clients before continuing the roll. For example:

- Manually relocate CAA services to the lead member. For example, to relocate the application resource named `cluster_lockd` to lead member `provolone`:

  ```
  # caa_relocate cluster_lockd -c provolone
  ```

- Temporarily modify the default cluster alias selection priority attribute, `selp`, to force the lead member to serve all client requests directed to that alias. For example:

  ```
  # cluamgr -a alias=DEFAULTALIAS,selp=100
  ```

  The lead member is now the end recipient for all connection requests and packets addressed to the default cluster alias.

  From another member or from an outside client, use services such as `telnet` and `ftp` to verify that the lead member can handle alias traffic. Test client access to all important services that the cluster provides.

  When you are satisfied, reset the alias attributes on the lead member to their original values.

7. Perform the postinstall stage (Section 4.8.5).

   On the lead member, run:

   ```
   # clu_upgrade postinstall
   ```

8. Perform the roll stage (Section 4.8.6).

   Roll the members of the cluster that have not already rolled.[1]

   You can roll multiple members simultaneously (parallel roll), subject to the restriction that the number of members not being rolled (plus the quorum disk, if one is configured) is sufficient to maintain cluster quorum.

   To roll a member, do the following:

   a. Halt the member system and boot it to single-user mode. For example:

      ```
      # shutdown -h now
      >>> boot -fl s
      ```

   b. When the system reaches single-user mode, run the following commands:

      ```
      # init s
      # bcheckrc
      # lmf reset
      ```

   c. Roll the member:

      ```
      # clu_upgrade roll
      ```

      If you are performing parallel rolls, use the `-f` option with the `clu_upgrade roll` command. This option causes the member to automatically reboot without first prompting for permission:

[1] The lead member was rolled during the install stage. Therefore, you do not perform the roll stage on the lead member.

```
# clu_upgrade -f roll
```

The roll command verifies that rolling the member will not result in a loss of quorum. If a loss of quorum will result, then the roll of the member does not occur and an error message is displayed. You can roll the member later, after one of the currently rolling members has rejoined the cluster and its quorum vote is available.

If the roll proceeds, the member is prepared for a reboot. If you used the `-f` option, no prompt is displayed; the reboot occurs automatically. If you did not use the `-f` option, `clu_upgrade` displays a prompt that asks whether you want to reboot at this time. Unless you want to examine something specific before you reboot, enter **`yes`**. (If you enter **`yes`**, it may take approximately half a minute before the actual reboot occurs.)

Perform parallel rolls to minimize the time needed to complete the roll stage. For example, on an eight-member cluster with a quorum disk, after rolling the lead member, you can roll four members in parallel.

i. Begin the roll stage on a member. (The lead member was rolled during the install stage. You do not perform the roll stage on the lead member.)

ii. When you see a message similar to the following, begin the roll stage on the next member:

```
   *** Info ***
You may now begin the roll of another cluster member.
```

If you see a message that begins like the following, it is probably caused by the number of currently rolling members that contribute member votes.

```
  *** Info ***
The current quorum conditions indicate that beginning
a roll of another member at this time may result in
the loss of quorum.
```

In this case, you have the following options:

- You can wait until a member completes the roll stage before you begin to roll the next member.
- If there is an unrolled member that does not contribute member votes, you can begin the roll stage on it.

d. Continue to roll members until all members of the cluster have rolled. Before starting each roll stage, wait until you see the message that it is all right to do so.

When you roll the last member, you will see a message similar to the following:

```
  *** Info ***
This is the last member requiring a roll.
```

Note

The roll actually takes place during the reboot. The `clu_upgrade roll` command sets up the `it`(8) scripts that will be run during the reboot. When you reboot, the `it` scripts roll the member, build a customized kernel, and then reboot again so the member will be running on its new customized kernel. When the member boots its

new customized kernel, it has completed its roll and is no longer running on tagged files.

---

9. Perform the switch stage (Section 4.8.7).

   After all members have rolled, run the `switch` command on any member.

   # **clu_upgrade switch**

10. One at a time, reboot each member of the cluster.

11. Perform the clean stage (Section 4.8.8).

    Run the following command on any member to remove the tagged (`.Old..`) files from the cluster and complete the upgrade.

    # **clu_upgrade clean**

4.4 Removing Patches Installed During a Rolling Upgrade

The following sections provide important information you need to be aware of if you remove or reinstall patches during a rolling upgrade.

4.4.1 Caution on Removing Version Switched Patches

When removing version switched patches on a cluster, do not remove version switched patches that were successfully installed in a previous rolling upgrade.

This situation can occur because more than one patch subset may contain the same version switched patch. Although both the new and old patches can be removed during a roll, only the most recently installed, newer version switched patch can be properly removed.

The older version switched patch can only be properly removed according to the documented procedure associated with that patch. This usually requires running some program before beginning the rolling upgrade to remove the patch.

If you accidentally remove the older version switched patch, the rolling upgrade will most likely fail on the switch stage. To correct this situation, you will have to undo the upgrade by undoing all the stages up to and including the "install" stage. You will then need to reinstall the original version switched patch from the original patch kit that contained it.

4.4.2 Steps Prior to the Switch Stage

You can remove a patch kit you installed during the rolling upgrade at any time prior to issuing the `clu_upgrade switch` command by returning to the install stage, rerunning `dupatch`, and selecting the `Patch Deletion` item in the Main Menu. See Section 3.8 for information about removing patches with `dupatch`.

The procedure is as follows:

1. Uninstall the patch kit as described in Section 3.8.

2. Run the `clu_upgrade undo install` command.

   Note that although you do not have to run the `clu_upgrade install` command when installing a patch kit or an NHD kit, you must run the `clu_upgrade undo install` command if you want to remove those kits and undo the install stage. After you run the `clu_upgrade undo install`, you can continue undoing stages as described in Section 4.6.

4.4.3 Steps for After the Switch Stage

To remove patches after you have issued the `clu_upgrade switch` command, you will have to complete the current rolling upgrade procedure and then rerun the procedure from the beginning (starting with the setup stage).

When you run the install stage, you must bring down your system to single-user mode as described in steps 1 through 6 of Section 3.3.1. When you rerun `dupatch` (step 7), select the `Patch Deletion` item in the Main Menu. See Section 3.8 for information about removing patches with `dupatch`.

If the patch uses the version switch, you can still remove the patch, even after you have issued the `clu_upgrade` switch command. Do this as follows:

1. Complete the current rolling upgrade procedure.
2. Undo the patch that uses the version switch by following the instructions in the release note for that patch. Note that the last step to undo the patch will require a shutdown of the entire cluster.
3. Rerun the rolling upgrade procedure from the beginning (starting with the setup stage). When you rerun `dupatch`, select the `Patch Deletion` item in the Main Menu.

Use the `grep` command to learn which patches use the version switch. For example, in the C shell:

```
# grep -l PATCH_REQUIRES_VERSION_SWITCH=\"Y\" /usr/.smdb./*PAT*.ctrl
```

For information about version switches, see Section 4.10.

Note

If you rerun the rolling upgrade procedure to remove patches, the prompts you receive during the setup stage will be different from those issued during the initial rolling upgrade. Those prompts will look as follows:

```
Do you want to continue to upgrade the cluster? [yes]: Return

What type of upgrade will be performed?

1) Rolling upgrade using the installupdate command
2) Rolling patch using the dupatch command
3) Both a rolling upgrade and a rolling patch
4) Exit cluster software upgrade

Enter your choice: 2
```

4.5 Displaying the Status of a Rolling Upgrade

The `clu_upgrade` command provides the following options for displaying the status of a rolling upgrade. You can run status commands at any time.

- To display the overall status of a rolling upgrade: `clu_upgrade -v` or `clu_upgrade -v status`.
- To determine whether you can run a stage: `clu_upgrade check [`*`stage`*`]`. If you do not specify a *`stage`*, `clu_upgrade` tests whether the next stage can be run.
- To determine whether a stage has started or completed: `clu_upgrade started` *`stage`* or `clu_upgrade completed` *`stage`*.
- To determine whether a member has rolled: `clu_upgrade check roll` *`memberid`*.

- To verify whether tagged files have been created for a layered product: `clu_upgrade tagged check [prod_code [prod_code ...]]`. If you do not specify a product code, `clu_upgrade` inspects all tagged files in the cluster.

Notes

During a roll, there might be two versions of the `clu_upgrade` command in the cluster — an older version used by members that have not yet rolled, and a newer version (if included in the update distribution or patch kit). The information that is displayed by the `status` command might differ depending on whether the command is run on a member that has rolled. Therefore, if you run the `status` command on two members, do not be surprised if the format of the displayed output is not the same.

If you run `clu_upgrade status` after running `installupdate`, `clu_upgrade` will display a message indicating that the install stage is complete. However, the install stage is not really complete until you run the `clu_upgrade postinstall` command.

4.6 Undoing a Stage

The `clu_upgrade undo` command provides the ability to undo a rolling upgrade that has not completed the switch stage. You can undo any stage except the switch stage and the clean stage. You must undo stages in order; for example, if you decide to undo a rolling upgrade after completing the preinstall stage, you undo the preinstall stage and then undo the setup stage.

Note

Before undoing any stage, we recommend that you read the relevant version of the *Cluster Release Notes* to determine whether there are restrictions related to the undoing of any stage.

To undo a stage, use the `undo` command with the stage that you want to undo. The `clu_upgrade` command determines whether the specified stage is a valid stage to undo. Table 4-3 outlines the requirements for undoing a stage:

Table 4-3: Undoing a Stage

Stage to Undo	Command	Comments
Setup	`clu_upgrade undo setup`	You must run this command on the lead member. In addition, no members can be running on tagged files when you undo the setup stage. Before you undo the setup stage, use the `clu_upgrade -v status` command to determine which members are running on tagged files. Then use the `clu_upgrade tagged disable` *`memberid`* command to disable tagged files on those members. (See Section 4.9 for information about tagged files and the commands used to manipulate them.) When no members are running on tagged files, run the `clu_upgrade undo setup` command on the lead member.
Preinstall	`clu_upgrade undo preinstall`	You must run this command on the lead member.

Table 4-3: Undoing a Stage (cont.)

Stage to Undo	Command	Comments
Install	`clu_upgrade undo install`	You can run this command on any member except the lead member. Halt the lead member. Then run the `clu_upgrade undo install` command on any member that has access to the halted lead member's boot disk. When the command completes, boot the lead member. If you installed a patch kit or individual patches during the install stage, you must first run `dupatch` to uninstall the patch kit before running the `clu_upgrade undo install` command. Section 4.4 describes the steps for removing a patch kit during a rolling upgrade.
Postinstall	`clu_upgrade undo postinstall`	You must run this command on the lead member.
Roll	`clu_upgrade undo roll` *`memberid`*	You can run this command on any member except the member whose roll stage will be undone. Halt the member whose roll stage is being undone. Then run the `clu_upgrade undo roll` *`memberid`* command on any other member that has access to the halted member's boot disk. When the command completes, boot the halted member. The member will now be using tagged files.

4.7 Rolling Upgrade Commands

The `clu_upgrade` command, described in `clu_upgrade`(8), controls the overall flow of a rolling upgrade and ensures that the stages are run in order. During the install stage, you run one or more of `installupdate`, `dupatch`, or `nhd_install` to load and install software. These commands are rolling upgrade aware; they are modified to understand which actions they are allowed to take during the install and roll stages of a rolling upgrade.

When you start a rolling upgrade, the cluster is running the software from the previous release. For the first part of any rolling upgrade, you are running the `clu_upgrade` command that is already installed on the cluster. If a new version is installed during the rolling upgrade, there may be minor differences in the on-screen display and behavior between the two versions of the command.

The following two tables show at which stages during a rolling upgrade new versions of upgrade commands, if shipped with the kits being installed, become available during a rolling upgrade:[2]

- Table 4-4 maps commands to stages for a rolling upgrade from Version 5.1A to Version 5.1B, a patch kit, or an NHD kit; or to Version 5.1B of the base operating system and cluster software followed by a patch of the new software within the same rolling upgrade.
- Table 4-5 maps commands to stages for a rolling upgrade from Version 5.1B to the next release of the operating system and cluster software, a Version 5.1B patch kit, or an NHD kit; or to the next release of the base operating system and cluster software followed by a patch of the new software within the same rolling upgrade.

[2] The `clu_upgrade version` command displays the version number for `clu_upgrade`. The `clu_upgrade` version numbers do not correspond with the version numbers of the operating system.

Table 4-4: Stages and clu_upgrade Versions When Performing a Rolling Upgrade from Version 5.1A

Stage	Version 5.1A	Next Release[a]	Comments
Preparation	X		The currently installed (old) version of `clu_upgrade` is always run in this stage.
Setup	X		The currently installed (old) version of `clu_upgrade` is always run in this stage. If performing an update installation, the new version of the `clu_upgrade` is extracted from the TruCluster software kit and installed at `/usr/sbin/clu_upgrade`, replacing the old version. Because this replacement is done before tagged files are created, all members will use the new `clu_upgrade` throughout the remainder of the rolling upgrade.
Preinstall		X	If the rolling upgrade includes an update installation, all members use the new version of `clu_upgrade` installed during the setup stage. (Otherwise, members continue to run the current version of `clu_upgrade`.)
Install		X	If the rolling upgrade includes an update installation, all members use the version of `clu_upgrade` installed during the setup stage. During the update installation, a new version of `installupdate` replaces the old one. A patch kit always installs the latest version of `dupatch`. If performing a patch, and if the patch kit includes a new version of `clu_upgrade`, the new version is installed and will be used by all cluster members starting with the postinstall stage.
Postinstall		X	If a new version of `clu_upgrade` was installed in either the setup stage or the install stage, all members use the new version.
Roll		X	If a new version of `clu_upgrade` was installed in either the setup stage or the install stage, all members use the new version.
Switch		X	If a new version of `clu_upgrade` was installed in either the setup stage or the install stage, all members use the new version.
Clean		X	If a new version of `clu_upgrade` was installed in either the setup stage or the install stage, all members use the new version.

[a] Version 5.1B of Tru64 UNIX and TruCluster software, a patch kit for Version 5.1A, or the installation of an NHD kit on Version 5.1A.

Table 4-5: Stages and clu_upgrade Versions When Performing a Rolling Upgrade from Version 5.1B

Stage	Version 5.1B	Next Release[a]	Comments
Preparation	X		The currently installed (old) version of `clu_upgrade` is always run in this stage.

Table 4-5: Stages and clu_upgrade Versions When Performing a Rolling Upgrade from Version 5.1B (cont.)

Stage	Version 5.1B	Next Release[a]	Comments
Setup	X		The currently installed (old) version of `clu_upgrade` is always run in this stage. If performing an update installation, the new version of the `clu_upgrade` is extracted from the TruCluster software kit and installed at `/usr/sbin/clu_upgrade`, replacing the old version. Because this replacement is done before tagged files are created, all members will use the new `clu_upgrade` throughout the remainder of the rolling upgrade.
Preinstall		X	If the rolling upgrade includes an update installation, all members use the new version of `clu_upgrade` installed during the setup stage. (Otherwise, members continue to run the current version of `clu_upgrade`.)
Install		X	If the rolling upgrade includes an update installation, all members use the version of `clu_upgrade` installed during the setup stage. During the update installation, a new version of `installupdate` replaces the old one. A patch kit always installs the latest version of `dupatch`. If performing a patch, and if the patch kit includes a new version of `clu_upgrade`, the new version is installed and will be used by all cluster members starting with the postinstall stage.
Postinstall		X	If a new version of `clu_upgrade` was installed in either the setup stage or the install stage, all members use the new version.
Roll		X	If a new version of `clu_upgrade` was installed in either the setup stage or the install stage, all members use the new version.
Switch		X	If a new version of `clu_upgrade` was installed in either the setup stage or the install stage, all members use the new version.
Clean		X	If a new version of `clu_upgrade` was installed in either the setup stage or the install stage, all members use the new version.

[a] The next release of Tru64 UNIX and TruCluster software, a patch kit for Version 5.1B, or the installation of an NHD kit on Version 5.1B.

4.8 Rolling Upgrade Stages

The following sections describe each of the rolling upgrade stages.

Note

These sections only describe the stages. Use the procedure in Section 4.3 to perform a rolling upgrade.

- Preparation stage (Section 4.8.1)
- Setup stage (Section 4.8.2)
- Preinstall stage (Section 4.8.3)

- Install stage (Section 4.8.4)
- Postinstall stage (Section 4.8.5)
- Roll stage (Section 4.8.6)
- Switch stage (Section 4.8.7)
- Clean stage (Section 4.8.8)

4.8.1 Preparation Stage

Command	Where Run	Run Level
`clu_upgrade -v check setup` *`lead_memberid`*	any member	multiuser mode

During the preparation stage, you back up all important cluster data and verify that the cluster is ready for a roll. Before beginning a rolling upgrade, do the following:

1. Choose one member of the cluster as the first member to roll. This member, known as the lead member, must have direct access to the root (`/`), `/usr`, `/var`, and, if used, `i18n` file systems.

   Make sure that the lead member can run any critical applications. You can test these applications after you update this member during the install stage, but before you roll any other members. If a problem occurs, you can try to resolve it on this member before you continue. If you cannot resolve a problem, you can undo the rolling upgrade and return the cluster to its pre-roll state. (Section 4.6 describes how to undo rolling upgrade stages.)

2. Back up the clusterwide root (`/`), `/usr`, and `/var` file systems, including all member-specific files in these file systems. If the cluster has a separate `i18n` file system, back up that file system. In addition, back up any other file systems that contain critical user or application data.

   Note

   If you perform an incremental or full backup of the cluster during a rolling upgrade, make sure to perform the backup on a member that is not running on tagged files. If you back up from a member that is using tagged files, you will only back up the contents of the `.Old..` files. Because the lead member never uses tagged files, you can back up the cluster from the lead member (or any other member that has rolled) during a rolling upgrade.

   Most sites have automated backup procedures. If you know that an automatic backup will take place while the cluster is in the middle of a rolling upgrade, make sure that backups are done on the lead member or on a member that has rolled.

3. If you plan to run the `installupdate` command in the install stage, remove any blocking layered products listed in Table 4-6 that are installed on the cluster.

4. Run the `clu_upgrade -v check setup` *`lead_memberid`* command, which verifies the following information:

   - No rolling upgrade is in progress.
   - All members are running the same versions of the base operating system and cluster software.
   - No members are running on tagged files.

- There is adequate free disk space.

5. Verify that each system's firmware will support the new software. Update firmware as needed before starting the rolling upgrade.

A cluster can continue to operate during a rolling upgrade because two copies exist of the operating system and cluster software files. (Only one copy exists of shared configuration files so that changes made by any member are visible to all members.) This approach makes it possible to run two different versions of the base operating system and the cluster software at the same time in the same cluster. The trade-off is that, before you start an upgrade, you must make sure that there is adequate free space in each of the clusterwide root (/), /usr, and /var file systems, and, if a separate domain exists for the Worldwide Language Support (WLS) subsets, in the i18n file system.

A rolling upgrade has the following disk space requirements:

- At least 50 percent free space in root (/), cluster_root#root.
- At least 50 percent free space in /usr, cluster_usr#usr.
- At least 50 percent free space in /var, cluster_var#var, plus, if updating the operating system, an additional 425 MB to hold the subsets for the new version of the base operating system.
- If a separate i18n domain exists for the WLS subsets, at least 50 percent free space in that domain.
- No tagged files are placed on member boot partitions. However, programs might need free space when moving kernels to boot partitions. We recommend that you reserve at least 50 MB free space on each member's boot partition.

Note

You cannot use the addvol command to add volumes to a member's root domain (the a partition on the member's boot disk). Instead, you must delete the member from the cluster, use diskconfig or SysMan to configure the disk appropriately, and then add the member back into the cluster.

- See the *Patch Summary and Release Notes* that came with your patch kit to find the amount of space you will need to install that kit. If installing an NHD kit, see the *New Hardware Delivery Release Notes and Installation Instructions* that came with your NHD kit to find the amount of space you will need to install that kit.

If a file system needs more free space, use AdvFS utilities such as addvol to add volumes to domains as needed. For information on managing AdvFS domains, see the Tru64 UNIX *AdvFS Administration* manual. (The AdvFS Utilities require a separate license.) You can also expand the clusterwide root (/) domain.

Note

The clu_upgrade command verifies whether sufficient space exists at the start of a rolling upgrade. However, nothing prevents a cluster member from consuming disk space during a rolling upgrade, thus creating a situation where a later stage might not have enough disk space.

Disk space is dynamic. If you know that a member will be consuming disk space during a rolling upgrade, add additional space before you start the upgrade.

4.8.2 Setup Stage

Command	Where Run	Run Level
`clu_upgrade setup` *`lead_memberid`*	any member	multiuser mode

The setup stage performs the `clu_upgrade check setup` command, creates tagged files, and prepares the cluster for the roll.

The `clu_upgrade setup` *`lead_memberid`* command performs the following tasks:

- Creates the rolling upgrade log file, `/cluster/admin/clu_upgrade.log`.
- Makes the `-v check setup` tests listed in Section 4.8.1.
- Prompts you to indicate whether to perform an update installation, install a patch kit, install an NHD kit, or a combination thereof. The following example shows the menu displayed by the TruCluster software Version 5.1B `clu_upgrade` command:

```
What type of rolling upgrade will be performed?

Selection   Type of Upgrade
------------------------------------------------------------
   1      An upgrade using the installupdate command
   2      A patch using the dupatch command
   3      A new hardware delivery using the nhd_install command
   4      All of the above
   5      None of the above
   6      Help
   7      Display all options again
------------------------------------------------------------
Enter your Choices (for example, 1 2 2-3):
```

- If you specify an update installation, copies the relevant kits onto disk:
  - If performing an update installation, copies the cluster kit to `/var/adm/update/TruClusterKit` so that the kit will be available to the `installupdate` command during the install stage. (The `installupdate` command copies the operating system kit to `/var/adm/update/OSKit` during the install stage.) The `clu_upgrade` command prompts for the absolute pathname for the TruCluster software kit location. On a TruCluster software Version 5.1B cluster, when performing a rolling upgrade that includes an update installation, remember to mount the TruCluster software kit before running the `clu_upgrade setup` command.
  - On a TruCluster software Version 5.1B cluster, if performing an NHD installation, uses the `nhd_install` command to copy the NHD kit to `/var/adm/update/NHDKit`

Caution

The files in `/var/adm/update` are critical to the roll process. Do not remove or modify files in this directory. Doing so can cause a rolling upgrade to fail.

- Creates the mandatory set of tagged files for the `OSF` (base), `TCR` (cluster), and `IOS` (Worldwide Language Support) products.

Caution

If, for any reason, during an upgrade you need to create tagged files for a layered product, see Section 4.9.

- Sets the `sysconfigtab` variable `rolls_ver_lookup=1` on all members except the lead member. When `rolls_ver_lookup=1`, a member uses tagged

files. As a result, the lead member can upgrade while the remaining members run on the .Old.. files from the current release.

- Prompts you to reboot all cluster members except the lead member. When the setup command completes, reboot these members one at a time so that the cluster can maintain quorum. This reboot is required for each member that will use tagged files in the mixed-version cluster. When the reboots complete, all members except the lead member are running on tagged files.

4.8.3 Preinstall Stage

Command	Where Run	Run Level
clu_upgrade preinstall	lead member	multiuser mode

The purpose of the preinstall stage is to verify that the cluster is ready for the lead member to run one or more of the installupdate, dupatch, or nhd_install commands.

The clu_upgrade preinstall command performs the following tasks:

- Verifies that the command is being run on the lead member, that the lead member is not running on tagged files, and that any other cluster members that are up are running on tagged files.
- (Optional) Verifies that tagged files are present, that they match their product's inventory files, and that each tagged file's AdvFS property is set correctly. (This process can take a while, but not as long as it does to create the tagged files in the setup stage. Table 4-2 provides time estimates for each stage.)
- Makes on-disk backup copies of the lead member's member-specific files.

4.8.4 Install Stage

Command	Where Run	Run Level
installupdate	lead member	single-user mode
dupatch	lead member	single-user or multiuser mode
nhd_install	lead member	single-user mode

If your current cluster is running TruCluster software Version 5.1B or Version 5.1A, you can perform one of the tasks or combinations of tasks listed in Table 4-1.

The install stage starts when the clu_upgrade preinstall command completes, and continues until you run the clu_upgrade postinstall command.

Note

If you run clu_upgrade status after running installupdate, clu_upgrade displays a message indicating that the install stage is complete. However, the install stage is not really complete until you run the clu_upgrade postinstall command.

The lead member must be in single-user mode to run the installupdate command or the nhd_install command; single-user mode is recommended for the dupatch command. When taking the system to single-user mode, you must halt the system and then boot it to single-user mode.

When the system is in single-user mode, run the init s, bcheckrc, and lmf reset commands before you run the installupdate, dupatch, or nhd_install commands. See the Tru64 UNIX *Installation Guide*, the Tru64 UNIX and

TruCluster software, and the Tru64 UNIX *New Hardware Delivery Release Notes and Installation Instructions* for information on how to use these commands.

Notes

You can run the `dupatch` command multiple times in order to install multiple patches. Doing so may make isolating problems difficult if any arise after the patch process is completed and the cluster is in use.

During the install stage, you cannot run a `dupatch` command followed by an `installupdate` command. To patch the current software before you perform a rolling upgrade, you must perform two complete rolling upgrade operations: one to patch the current software, and one to perform the update installation.

If an NHD installation is part of a rolling upgrade that includes an update installation, you do not have to manually run `nhd_install`; the `installupdate` command will install the NHD kit. Otherwise, use the `nhd_install` command copied by `clu_upgrade` during the setup stage: `/var/adm/update/NHDKit/nhd_install`.

4.8.5 Postinstall Stage

Command	Where Run	Run Level
`clu_upgrade postinstall`	lead member	multiuser mode

The postinstall stage verifies that the lead member has completed an update installation, a patch, or an NHD installation. If an update installation was performed, `clu_upgrade postinstall` verifies that the lead member has rolled to the new version of the base operating system.

4.8.6 Roll Stage

Command	Where Run	Run Level
`clu_upgrade roll`	member being rolled	single-user mode

The lead member was upgraded in the install stage. The remaining members are upgraded in the roll stage.

In many cluster configurations, you can roll multiple members in parallel and shorten the time required to upgrade the cluster. The number of members rolled in parallel is limited only by the requirement that the members not being rolled (plus the quorum disk, if one is configured) have sufficient votes to maintain quorum. Parallel rolls can be performed only after the lead member is rolled.

The `clu_upgrade roll` command performs the following tasks:

- Verifies that the member is not the lead member, that the member has not already been rolled, and that the member is in single-user mode. Verifies that rolling the member will not result in a loss of quorum.
- Backs up the member's member-specific files.
- Sets up the `it`(8) scripts that will be run on reboot to perform the roll.
- Reboots the member. During this boot, the `it` scripts roll the member, build a customized kernel, and reboot with the customized kernel.

Note

If you need to add a member to the cluster during a rolling upgrade, you must add the member from a member that has completed its roll.

If a member goes down (and cannot be repaired and rebooted) before all members have rolled, you must delete the member to complete the roll of the cluster. However, if you have rolled all members but one, and this member goes down before it has rebooted in the roll stage, you must delete this member and then reboot any other member of the cluster. (The `clu_upgrade` command runs during reboot and tracks the number of members rolled versus the number of members currently in the cluster; `clu_upgrade` marks the roll stage as completed when the two values are equal. That is why, in the case where you have rolled all members except one, deleting the unrolled member and rebooting another member completes the roll stage and lets you continue the rolling upgrade.)

4.8.7 Switch Stage

Command	Where Run	Run Level
`clu_upgrade switch`	any member	multiuser mode All members must be up and running[a]

[a] You can override this requirement by using the `-f` option to the switch command. However, all members' boot disks must be accessible for the `-f` option to work.

The switch stage sets the active version of the software to the new version, which results in turning on any new features that had been deliberately disabled during the rolling upgrade. (See Section 4.10 for a description of active version and new version.)

The `clu_upgrade switch` command performs the following tasks:

- Verifies that all members have rolled, that all members are running the same versions of the base operating system and cluster software, and that no members are running on tagged files.
- Sets the new version ID in each member's `sysconfigtab` file and running kernel.
- Sets the active version to the new version for all cluster members.

Note

After the switch stage completes, you must reboot each member of the cluster, one at a time.

4.8.8 Clean Stage

Command	Where Run	Run Level
`clu_upgrade clean`	any member	multiuser mode

The clean stage removes the tagged (`.Old..`) files from the cluster and completes the upgrade.

The `clu_upgrade clean` command performs the following tasks:

- Verifies that the switch stage has completed, that all members are running the same versions of the base operating system and cluster software, and that no members are running on tagged files.
- Removes all `.Old..` files.

- Removes any on-disk backup archives that `clu_upgrade` created.
- If the directory exists, recursively deletes `/var/adm/update/TruClusterKit`, `/var/adm/update/OSKit`, and `/var/adm/update/NHDKit`.
- If an update installation was performed, gives you the option of running the Update Administration Utility (`updadmin`) to manage the files that were saved during an update installation.
- Creates an archive directory for this upgrade, `/cluster/admin/clu_upgrade/history/`*`release_version`*, and moves the `clu_upgrade.log` file to the archive directory.

4.9 Tagged Files

A rolling upgrade updates the software on one cluster member at a time. To support two versions of software within the cluster during a roll, `clu_upgrade` creates a set of tagged files in the setup stage.

A tagged file is a copy of a current file with `.Old..` prepended to the copy filename, and an AdvFS property (`DEC_VERSION_TAG`) set on the copy. For example, the tagged file for the `vdump` command is named `/sbin/.Old..vdump`. Because tagged files are created in the same file system as the original files, you must have adequate free disk space before beginning a rolling upgrade.

Whether a member is running on tagged files is controlled by that member's `sysconfigtab rolls_ver_lookup` variable. The upgrade commands set the value to `1` when a member must run on tagged files, and to `0` when a member must not run on tagged files.

If a member's `sysconfigtab rolls_ver_lookup` attribute is set to `1`, pathname resolution includes determining whether a specified filename has a `.Old..`*`filename`* copy and whether the copy has the `DEC_VERSION_TAG` property set on it. If both conditions are met, the requested file operation is transparently diverted to use the `.Old..`*`filename`* version of the file. Therefore, if the `vdump` command is issued on a member that has not rolled, the `/sbin/.Old..vdump` file is executed; if the command is issued on a member that has rolled, the `/sbin/vdump` file is executed. The only member that never runs on tagged files is the lead member (the first member to roll).

Note

File system operations on directories are not bound by this tagged file restraint. For example, an `ls` of a directory on any cluster member during a rolling upgrade lists both versions of a file. However, the output of an `ls -ail` command on a member that has not rolled is different from the output on a member that has rolled. In the following examples the `ls -ail` command is run first on a member that has not rolled and then on a member that has rolled. (The `awk` utility is used to print only the inode, size, month and day timestamp, and name of each file.)

The following output from the `ls` command is taken from a cluster member running with tags before it has rolled. The tagged files are the same as their untagged counterparts (same inode, size, and timestamp). When this member runs the `hostname` command, it runs the tagged version (inode 3643).

```
# cd /sbin
# ls -ail hostname .Old..hostname ls .Old..ls init .Old..init |\
awk '{printf("%d\t%d\t%s %s\t%s\n",$1,$6,$7,$8,$10)}'

 3643   16416    Aug 24   .Old..hostname
 3648   395600   Aug 24   .Old..init
 3756   624320   Aug 24   .Old..ls
```

```
 3643   16416    Aug 24   hostname
 3648   395600   Aug 24   init
 3756   624320   Aug 24   ls
```

The following output from the `ls` command is taken from a cluster member running without tags after it has rolled. The tagged files now differ from their untagged counterparts (different inode, size, and timestamp). When this member runs the `hostname` command, it runs the non-tagged version (inode 1370).

```
# cd /sbin
# ls -ail hostname .Old..hostname ls .Old..ls init .Old..init |\
awk '{printf("%d\t%d\t%s %s\t%s\n",$1,$6,$7,$8,$10)}'

 3643   16416    Aug 24   .Old..hostname
 3648   395600   Aug 24   .Old..init
 3756   624320   Aug 24   .Old..ls
 1187   16528    Mar 12   hostname
 1370   429280   Mar 12   init
 1273   792640   Mar 12   ls
```

After you create tagged files in the setup stage, we recommend that you run any administrative command, such as `tar`, from a member that has rolled. You can always run commands on the lead member because it never runs on tagged files.

The following rules determine which files have tagged files automatically created for them in the setup stage:

- Tagged files are created for inventory files for the following product codes: base operating system (`OSF`), TruCluster software (`TCR`), and Worldwide Language Support (`IOS`). (The subsets for each product use that product's three-letter product code as a prefix for each subset name. For example, TruCluster software subset names start with the TruCluster software three-letter product code: `TCRBASE510`, `TCRMAN510`, and `TCRMIGRATE510`.)
- By default, files that are associated with other layered products do not have tagged files created for them. Tagged files are created only for layered products that have been modified to support tagged files during a rolling upgrade.

Caution

Unless a layered product's documentation specifically states that you can install a newer version of the product on the first rolled member, and that the layered product knows what actions to take in a mixed-version cluster, we strongly recommend that you do not install either a new layered product or a new version of a currently installed layered product during a rolling upgrade.

The `clu_upgrade` command provides several `tagged` command options to manipulate tagged files: `check`, `add`, `remove`, `enable`, and `disable`. When dealing with tagged files, take the following into consideration:

- During a normal rolling upgrade you do not have to manually add or remove tagged files. The `clu_upgrade` command calls the `tagged` commands as needed to control the creation and removal of tagged files.
- If you run a `clu_upgrade tagged` command, run the `check`, `add`, and `remove` commands on a member that is not running on tagged files; for example, the lead member. You can run the `disable` and `enable` commands on any member.
- The target for a `check`, `add`, or `remove` tagged file operation is a product code that represents an entire product. The `clu_upgrade tagged` commands operate on all inventory files for the specified product or products. For example, the following command verifies the correctness of all the tagged files created for the `TCR` kernel layered product (the TruCluster software subsets):

```
# clu_upgrade tagged check TCR
```

If you inadvertently remove a `.Old..` copy of a file, you must create tagged files for the entire layered product to re-create that one file. For example, the `vdump` command is in the `OSFADVFS`*xxx* subset, which is part of the `OSF` product. If you mistakenly remove `/sbin/.Old..vdump`, run the following command to re-create tagged files for the entire layered product:

```
# clu_upgrade tagged add OSF
```

- The `enable` and `disable` commands enable or disable the use of tagged files by a cluster member. You do not have to use `enable` or `disable` during a normal rolling upgrade.

  The `disable` command is useful if you have to undo the setup stage. Because no members can be running with tagged files when undoing the setup stage, you can use the `disable` command to disable tagged files on any cluster member that is currently running on tagged files. For example, to disable tagged files for a member whose ID is 3:

  ```
  # clu_upgrade tagged disable 3
  ```

  The `enable` command is provided in case you make a mistake with the `disable` command.

4.10 Version Switch

A version switch manages the transition of the active version to the new version of an operating system. The active version is the one that is currently in use. The purpose of a version switch in a cluster is to prevent the introduction of potentially incompatible new features until all members have been updated. For example, if a new version introduces a change to a kernel structure that is incompatible with the current structure, you do not want cluster members to use the new structure until all members have updated to the version that supports it.

At the start of a rolling upgrade, each member's active version is the same as its new version. When a member rolls, its new version is updated. After all members have rolled, the switch stage sets the active version to the new version on all members. At the completion of the upgrade, all members' active versions are again the same as their new versions. The following simple example uses an active version of `1` and a new version of `2` to illustrate the version transitions during a rolling upgrade:

```
All members at start of roll:   active (1)  = new (1)
Each member after its roll:     active (1) != new (2)
All members after switch stage: active (2)  = new (2)
```

The `clu_upgrade` command uses the `versw` command, which is described in `versw`(8), to manage version transitions. The `clu_upgrade` command manages all the version switch activity when rolling individual members. In the switch stage, after all members have rolled, the following command completes the transition to the new software:

```
# clu_upgrade switch
```

4.11 Rolling Upgrade and Layered Products

This section discusses the interaction of layered products and rolling upgrades:

- General guidelines (Section 4.11.1)
- Blocking layered products (Section 4.11.2)

4.11.1 General Guidelines

The `clu_upgrade setup` command prepares a cluster for a rolling upgrade of the operating system. Do not use the `setld` command to load software onto the cluster between performing the `clu_upgrade setup` command and rolling the first cluster member to the new version. If you install software between performing the `clu_upgrade setup` command and rolling a cluster member to the new version, the new files will not have been processed by `clu_upgrade setup`. As a result, when you roll the first cluster member, these new files will be overwritten.

If you must load software:

- Wait until at least one member has rolled.
- Install the software on a member that has rolled.

4.11.2 Blocking Layered Products

A blocking layered product is a product that prevents the `installupdate` command from completing. Blocking layered products must be removed from the cluster before starting a rolling upgrade that will include running the `installupdate` command. You do not have to remove blocking layered products when performing a rolling upgrade solely to patch the cluster or install an NHD kit.

Table 4-6 lists blocking layered products for this release.

Table 4-6: Blocking Layered Products

Product Code	Description
3X0	Open3D
4DT	Open3D
ATM	Atom Advanced Developers Kit
DCE	Distributed Computing Environment
DNA	DECnet
DTA	Developer's Toolkit (Program Analysis Tools)
DTC	Developer's Toolkit (C compiler)
MME	Multimedia Services
O3D	Open 3D
PRX	PanoramiX Advanced Developers Kit

Notes

The three-letter product codes are the first three letters of subset names. For example, a subset named ATMBASE*xxx* is part of the ATM product (Atom Advanced Developers Kit), which is a blocking layered product. However, a subset named OSFATMBIN*xxx* contains the letters ATM, but the subset is not part of a blocking layered product; it is a subset in the OSF product (the base operating system).

When a blocking layered product is removed as part of the rolling upgrade, it is removed for all members. Any services that rely on the blocking product will not be available until the roll completes and the blocking layered product is reinstalled.

4.12 Rolling Upgrade and RIS

When performing the install stage of a rolling upgrade, you can load the base operating system subsets from a CD-ROM or from a Remote Installation Services (RIS) server.

Note

You can use RIS only to load the base operating system subsets.

To use RIS, you must register both the lead member and the default cluster alias with the RIS server. When registering for operating system software, you must provide a hardware address for each host name. Therefore, you must create a hardware address for the default cluster alias in order to register the alias with the RIS server. (RIS will reject an address that is already in either of the RIS server's `/etc/bootptab` or `/var/adm/ris/clients/risdb` files.)

If your cluster uses the cluster alias virtual MAC (vMAC) feature, register that virtual hardware address with the RIS server as the default cluster alias's hardware address. If your cluster does not use the vMAC feature, you can still use the algorithm that is described in the vMAC section of the *Cluster Administration* manual to manually create a hardware address for the default cluster alias.

A vMAC address consists of a prefix (the default is AA:01) followed by the IP address of the alias in hexadecimal format. For example, the default vMAC address for the default cluster alias `deli` whose IP address is `16.140.112.209` is `AA:01:10:8C:70:D1`. The address is derived in the following manner:

```
Default vMAC prefix:       AA:01
Cluster Alias IP Address:  16.140.112.209
IP address in hex. format: 10.8C.70.D1
vMAC for this alias:       AA:01:10:8C:70:D1
```

Another method for creating a hardware address is to append an arbitrary string of eight hexadecimal numbers to the default vMAC prefix, `AA:01`. For example, `AA:01:00:00:00:00`. Make sure that the address is unique within the area served by the RIS server. If you have more than one cluster, remember to increment the arbitrary hexadecimal string when adding the next alias. (The vMAC algorithm is useful because it creates an address that has a high probability of being unique within your network.)

5

No-Roll Patching

The no-roll patch process lets you install patches on a cluster without performing a rolling upgrade. This chapter provides the following information:

- An overview of the no-roll patch process (Section 5.1)
- A step-by-step description of the process as it differs from a normal `dupatch` session (Section 5.2)
- Throwing the version switch (Section 5.3)
- How to remove patches from a cluster using the no-roll patch method (Section 5.4)

Note

The no-roll technology is included in Rev. 34–00 and higher of the `dupatch` utility. You can find the revision number on the first output line you see when you run `dupatch` (see the example in Section 5.2). The first kit that includes this technology was issued in April 2002.

5.1 Overview

A rolling upgrade lets you perform a software upgrade on a cluster while maintaining high availability of the cluster. To provide this high availability, a certain amount of setup work is required to build tagged files and to reboot the cluster members to use the tagged files. This can take a considerable amount of time.

However, if you have a mission-critical environment and want to use a patch method that applies patches quickly, minimizes down time of the cluster, and reduces the number of reboots required, you might want to use the no-roll patch process. This process patches your cluster in one operation that requires only one or two reboots of the whole cluster to complete the operation. You will need the second reboot only if you install a patch that contains a version switch (see Section 5.3).

The no-roll patch process is a modification of `dupatch`; that is, all patches are installed or removed entirely using the `dupatch` utility, as opposed to the `clu_upgrade` and `dupatch` utilities used in the rolling upgrade procedure. The no-roll process conducts significantly fewer operations than the rolling upgrade procedure.

While a no-roll patch installation is in progress, no other critical operations should be running on the cluster because the cluster will change state and reboot automatically at various stages of the procedure.

In addition, the no-roll patch procedure employs the use of the Tru64 UNIX Event Management System (EVM) to send cluster-wide events. As a result, patches must be applied to the system in multiuser mode. If you attempt to use the no-roll procedure while in single-user mode, you will be advised to change the cluster to multiuser mode before continuing.

5.2 Steps for Running a No-Roll Procedure

The following steps describe how to patch your cluster using the no-roll procedure.

NOTE

To use the no-roll patch method, you must not use the `clu_upgrade` utility to prepare the cluster, as you would for a rolling upgrade prior to running `dupatch`. If a rolling upgrade is in progress before attempting to run `dupatch`, then the no-roll option will not be available until the cluster is restored to the state prior to the roll attempt.

1. With your system running in multiuser mode, enter the `dupatch` command:

```
# dupatch
Tru64 UNIX Patch Utility (Rev. 48-00)
==========================
This dupatch session is logged in /var/adm/patch/log/session.log

    Main Menu:
    ---------

    1)  Patch Kit Installation
    2)  Patch Kit Deletion
    3)  Patch Kit Documentation

    4)  Patch Tracking
    5)  Patch Baseline Analysis/Adjustment

    h)  Help on Command Line Interface

    q)  Quit

Enter your choice:
```

2. From the main menu select the patch installation or patch deletion option. (See Section 3.3.2.)
3. If `dupatch` determines it is running on a cluster that has not been prepared to do a rolling patch, it asks if you want to do the patch operation without rolling. You will see a message similar to the following:

```
Checking Cluster State...done
This system is part of a cluster which has not been prepared to do a
rolling patch installation or deletion. Do you wish to perform this
patch operation cluster-wide without using the rolling-patch mechanism?

Please answer y or n ? [y/n]:
```

   If you choose `y`, `dupatch` proceeds by allowing you to do the analysis and selection of patches to be installed or removed, after which the whole cluster is brought down to `init` level 2 via an Event Management System event.

   If you are using `dupatch` from the command line and do not specify the `-proceed` option, you will need to press Return in order to transition the cluster from level 3 to level 2. If the `-proceed` option was set, the transition will occur automatically.

After `dupatch` completes its patch analysis, it will perform the patch operation on the member on which you ran `dupatch`. After the patches are installed or removed, `dupatch` will issue a second event to the remaining cluster members that will instruct them to complete their patch operations in parallel.

The `dupatch` utility then waits a calculated time-out period for all the other cluster members to complete their operations. The time-out period is based on the time it took to perform the patch operation on the member running `dupatch`.

After the patch operation is completed on all other cluster members, `dupatch` will complete the procedure on the member on which the `dupatch` command was issued.

If a cluster member times out or encounters an error, dupatch will report the problem, suspend the process, and send you a message to check the problematic member in order to resolve the problem. Once dupatch has resumed, it will complete the patch process on the rest of the cluster.

If a cluster member is known to be down when you issue the dupatch command, an /sbin/it job will be posted for the member to run the cluster patch script upon reboot. (For more information, see the it(8) reference page.)

Because all patches currently require a reboot, the whole cluster will reboot after all the members report back.

5.3 Throwing the Version Switch

If a patch applied to the system requires the use of a version switch, you will see a message similar to the following at the end of the dupatch session:

```
**********************************************************************
Patch OSFPAT00074200510 has been
identified as needing a version switch. Once the following reboot is
complete, please enter the "/var/adm/patch/noroll/noroll_versw"
command from any cluster member.
**********************************************************************
```

As indicated by the message, you must enter the /var/adm/patch/noroll/noroll_versw command from any cluster member. This is a manual operation that you must perform after the reboot is complete. All cluster members must be up prior to running the noroll_versw command. If they are not, the noroll_versw command will fail and the version switch will not take place.

After issuing the noroll_versw command, reboot your system to ensure system integrity.

5.4 Removing Patches

You cannot use the no-roll process to remove inclusive patch kits because you must run the versw_enable_delete script (Section 3.8.2.1), which requires that you reboot each cluster member to remove the patch kit. Because the no-roll process automatically reboots the system after deleting the patches, you would not be able to reboot each member as required.

A

Viewing Log files

The dupatch utility captures patching activities in the following log files:

- /var/adm/patch/log/session.log

  Every time you run dupatch it creates a session log that captures dupatch activities. The session.log files from the previous 25 sessions are saved. The order is first in, first out, with session.log.25 as the oldest file.

- /var/adm/patch/log/Dupatch_load_*Date*.log

  When you run dupatch from the newly untarred kit or from the mounted Tru64 UNIX Patch CD–ROM, dupatch determines if the patch distribution contains new patch tools, and loads them if necessary.

  This log file has a name similar to this:
  Dupatch_load_2000Jul1:15:43:35.log

- /var/adm/patch/log/baseline.log

  When you run the system baselining feature, dupatch creates a baseline log. The session.log files from the previous 25 sessions are saved. The order is first in, first out, with baseline.log.25 as the oldest file.

- /var/adm/patch/log/event.log

  When patches are installed or removed, an event log captures that information. Only one copy of the file is updated each time patches are installed or removed. The information in the patch event log is not available through the dupatch user interface, but the log is a text file that you can view with a command such as more. The following list describes the types of information an event log provides, although the format and content are subject to change. Example A-1 shows a typical event log.

DUPATCH_REV>	The revision of dupatch being used
TYPE>	The type of action that was taken; either install or remove
NAME>	The name entered by the user through a dupatch query
USER>	The name of the user performing the action
NOTES>	Notes that were entered by the user through a dupatch query
KITLOC>	The directory from which the patch kit was installed
KITNAME>	The name of the patch kit that was installed
REVERT>	The choice made on whether or not the patch installation is reversible
BACKUP_DIRECTORY>	A pointer to the directory that contains the original files before they were patched
BACKUP_SETUP>	A plain directory; not a mount point or a symbolic link
SUCCEED>	A list of patches for which the action succeeded
FAIL>	A list of patches for which the action failed

Example A-1: Sample Event Log

```
<RECORD>
DUPATCH_REV>30-01
TYPE>install
NAME>mstone
USER>mstone
DATE>Mon Jul 3 13:03:33 EST 2000
NOTES>Install BL13 patches from CD-ROM
>
KITLOC>/cdrom/DIGITAL_UNIX_V4.0F/patch_kit/DIGITAL_UNIX_V4.0F/kit
KITNAME><DUV40FAS0004-20000613> OSF440
REVERT>Y
BACKUP_DIRECTORY>//var/adm/patch/backup
BACKUP_SETUP>
SUCCEED>OSFPAT00001900440
```

B

Common Error, Warning, and Informational Messages

This appendix describes error, warning, and informational messages for the dupatch utility. The following information is provided for each message:

Source: The function that generates the message.

Problem: A brief description of possible causes for the message.

Causes: A summary of situations that cause the message.

Action: General recovery guidance.

Output: A sample of the message.

B.1 Patch Preinstallation Check and Installation Messages

The following sections describe messages you might see when running the dupatch preinstallation check or installation functions.

B.1.1 Patch Installation Blocked by Unknown System File

Source: dupatch preinstallation check or installation.

Problem: The installation of a specific patch is blocked due to an existing system file that is unknown.

Cause: This situation usually occurs when system files are placed on the system through manual intervention. For example, this may have been the result of installing a Customer-Specific patch received from HP Services or a system administrator's customization of a Tru64 UNIX file.

Until you confirm otherwise, the unknown system files should be viewed as intentional customizations that are important for proper system operation. As such, care should be taken to understand why the system files have been customized.

Action: Determine the origin of the existing unknown system files. The steps you take will be determined by the reason your system files were manually changed. See Section 2.2 for more information.

Output:

```
Checking patch prerequisites and patch file applicability ...
       (depending upon the number of patches you select, this may take a while)
 ------------------------------------------------------------------------
     Problem installing:

      - DIGITAL_UNIX_V4.0F / Common Desktop Environment (CDE) Patches:

             Patch 0326.00 - CDE Login Correction

             ./usr/dt/bin/dtwm:
                     its origin cannot be identified.

     This patch will not be installed.
  ------------------------------------------------------------------------
       * Following patch(es) failed in prerequisite/file applicability check:

      - TRU64_UNIX_V4.0D / Common Desktop Environment (CDE) Patches:
             Patch 0326.00 - CDE Login Correction
```

B.1.2 Patch Installation Blocked by Missing System File

Source: dupatch preinstallation check or installation.

Problem: Installation of a specific patch is blocked due to missing system file.

Causes: This situation usually occurs when a system file that was installed with setld is manually removed from the system. The file is marked as installed in the system inventory records.

Action: Determine why the system file is missing and whether it is safe to enable dupatch to install the blocked patch. See Section 2.2 for more information.

Output:

```
Checking patch prerequisites and patch file applicability...
  (depending upon the number of patches you select, this may take a while)
------------------------------------------------------------------------

Problem installing:

 - DIGITAL_UNIX_V4.0F / Commands, Shells, & Utility Patches:
        Patch 0236.00 - vi Editor Correction

   ./usr/bin/vedit:
           does not exist on your system,
           however, it is in the inventory of installed subsets.

 This patch will not be installed.

 ------------------------------------------------------------------------
    * Following patch(es) failed in prerequisite/file applicability check:

  - DIGITAL_UNIX_V4.0F / Commands, Shells, & Utility Patches:
         Patch 0236.00 - vi Editor Correction
```

B.1.3 Installation Blocked by Layered Product Collision

Source: dupatch preinstallation check or installation.

Problem: The installation of a specific patch is blocked due to an existing system file that is installed by a layered product.

Causes: A small set of layered products deliver updated Tru64 UNIX operating system files.

Action: To resolve this situation contact the Product Customer Services representative.

Output:

```
Checking patch prerequisites and patch file applicability...
   (depending upon the number of patches you select, this may take a while)
 ------------------------------------------------------------------------

 Problem installing:

  - TRU_UNIX_V4.0F / Network Patches:
        Patch 0182.00 - xti/streams Interface Module Correction

        ./sys/BINARY/xtiso.mod:
                is installed by:

                                BLTLPCONFLICTTEST410

                and can not be replaced by this patch.

 This patch will not be installed.

 ------------------------------------------------------------------------
```

```
   * Following patch(es) failed in prerequisite/file applicability check:

 - DIGITAL_UNIX_V4.0F / Network Patches:
       Patch 0182.00 - xti/streams Interface Module Correction
```

B.1.4 Patch Installation Blocked by Dependencies on Other Patches

Source: dupatch preinstallation check or installation.

Problem: The installation of a specific patch is blocked due to its dependency on other uninstalled patches.

Causes: This usually occurs when you miss the selection of all dependent patches. It only occurs in old style patch kits.

Action: Through the dupatch Installation Menu, take one of the following actions:

- Reselect the patches including the noted dependent patch and attempt reinstallation; dupatch will notify you of other missing dependent patches.
- Select all patches and proceed with patch installation.

Output:

```
SAMPLE OUTPUT:

 Checking patch prerequisites and patch file applicability...
   (depending upon the number of patches you select, this may take a while)
 ------------------------------------------------------------------------

 Problem installing:

  - DIGITAL_UNIX_V4.0F / Security Related Patches:
       Patch 0579.01 - Security, Various Kernel Fixes (SSRT0482U)

 requires the existence of the following un-installed/un-selected subset(s):

  - TruCluster_V1.6 / Filesystem Patches:
       Patch 0037.00 - Support For New AdvFS Mount Option "-o noatimes"

  - TruCluster_V1.6 / ASE Availability Manager (AM) Patches:
       Patch 0033.00 - Kern Mem Fault And simple_lock Panic Correction

 This patch will not be installed.

 ------------------------------------------------------------------------
    * Following patch(es) failed in prerequisite/file applicability check:

  - TRU64L_UNIX_V4.0F / Security Related Patches:
       Patch 0579.01 - Security, Various Kernel Fixes (SSRT0482U)
```

B.1.5 Patch Installation Blocked by Missing Product Subset

Source: dupatch preinstallation check or installation.

Problem: A specific patch cannot be installed because the product software subset is not installed on your system.

Causes: This is usually an informational message and no further action is required. However, this message may also occur due to an internal patch kit error that results in an incorrectly specified patch dependency.

Action: If the specific patch being blocked is the only patch being blocked you can assume this is an informational message. It may be an internal patch kit error if there are other patches whose installation is blocked by the patch whose subset is not installed. As a workaround, if you need one of the other patches whose installation is blocked, you can install the optional Tru64 UNIX or TCR release subset and reinstall the patches.

Output:

```
Checking patch prerequisites and patch file applicability...
   (depending upon the number of patches you select, this may take a while)
 ------------------------------------------------------------------------

 Problem installing:

  - TruCluster_V1.6 / Cluster Kernel Patches:
         Patch 0035.00 - rm_spur Driver Correction

 requires the existence of the following un-installed/un-selected subset(s):

 - TruCluster_V1.6 - subset: TCRMCA141

 This patch will not be installed.

 ------------------------------------------------------------------------
   * Following patch(es) failed in prerequisite/file applicability check:

  - TruCluster_V1.6 / Cluster Kernel Patches:
         Patch 0035.00 - rm_spur Driver Correction
```

B.1.6 Patch Installation Blocked by Disk Space

Source: `dupatch` preinstallation check or installation.

Problem: The system disk did not have enough space to install patches.

Causes: This occurs when there is not enough disk space in `/`, `/var`, or `/usr` partitions for `dupatch` to archive the existing system files and move the patched files into place.

Action: Provide the necessary disk space and reinstall patches. If you cannot provide enough system disk space through other means, you may want to make `/var/adm/patch/backup` a symbolic link to or NFS-mount another file system that is not related to the `/`, `/var`, or `/usr` partitions.

Output:

```
Checking patch prerequisites once more...
       (depending upon the number of patches you select, this may take a while)

     ./usr/lbin/fitset:
     file system /whd needs 65829 Kbytes more to install the software specified.

            There is not enough file system space to install all the patches.
            you have selected.

            Please press RETURN to start another selection.
                      .
                      .
                      .
```

B.1.7 Patch Installation Blocked by Installed Patch or Subset

Source: `dupatch` preinstallation check or installation.

Problem: The patch you are trying to install is built so it cannot supersede the later revision patch or subset that is installed on your system.

Causes: This applicability feature is used to ensure that your system is not regressed through the installation of older code.

Action: If the situation is caused by a Release patch being blocked by a layered product or other subsets, contact your service provider.

Output:

```
Problem installing:

 - DIGITAL_UNIX_V4.0D / Filesystem Patches:
```

```
        Patch   00016.01 - System Run Level Correction

        ./sbin/.new..bcheckrc:
                is installed by:


 - DIGITAL_UNIX_V4.0D:
        Patch C 00484.01

                and can not be replaced by this patch.

This patch will not be installed.
```

B.1.8 Patch Installation Blocked by an Existing CSP

Source: dupatch preinstallation check or installation.

Problem: Release patches will not automatically supersede a Customer-Specific patch (CSP).

Causes: A file you are trying to update with a Release patch has been previously updated through the installation of a CSP. The Release patch does not have any knowlege as to whether it contains fixes contained in CSPs.

Action: Determine if the CSP is included in the Release Patch Kit:

- If yes, then you can safely remove the CSP (via dupatch) and reinstall the Release patch .
- If no, contact your service provider to determine how to proceed.

Output:

```
Problem installing:

 - DIGITAL_UNIX_V4.0F / Commands, Shells, & Utility Patches:
         Patch   00444.00 - Fixes sort problem when running in Japanese locale

         ./usr/bin/sort:
                is installed by Customer Specific Patch (CSP):

  - DIGITAL_UNIX_V4.0F:
         Patch C 00187.00

                and can not be replaced by this patch. To install this patch,
                you must first remove the CSP using dupatch. Before performing
                this action, you should contact your Service
                Representative to determine if this patch kit contains the
                CSP. If it does not, you may need to obtain a new CSP in order
                to install the patch kit and retain the CSP fix.
```

B.1.9 The dupatch Tools Are Outdated

Source: dupatch preinstallation check or installation.

Problem: Patch tool set residing on system are not the most recent version.

Causes: If the dupatch utility delivered with the patch kit determines that the tools residing on the system are not consistent with the patch kit, it will copy over updated versions of utilities used by dupatch.

Action: This is an informational message and no further action is required.

Output:

```
  Patch tools need to be installed or updated on your system.
 Please invoke the command as the super-user (root) first.

 * A new version of patch tools required for patch management
   is now being installed on your system.
```

B.1.10 Some Patches Must Be Made Reversible

Source: dupatch preinstallation check or installation.

Causes: The user tried to install a patch as nonreversible; however, the patch in question must be installed as reversible.

Action: This is an informational message and no further action is required.

Output:

```
* The following patch(es) are required to be reversible and
      will be made reversible automatically:

 - DIGITAL_UNIX_V4.0F / Commands, Shells, & Utility Patches:
        Patch C 00187.00 - v 4.0f patch E C187.00
```

B.2 Patch Removal Messages

The following sections describe messages you might see when running the dupatch patch deletion function.

B.2.1 Patch Removal Blocked by Missing Patch Backup Files

Source: dupatch deletion.

Problem: An attempt to remove a specific patch or all patches fails because the backup of the prepatch system files is not available to dupatch.

Causes: The /var/adm/patch/backup area does not contain the prepatch system files.

Action: Ensure that dupatch can access the/var/adm/patch/backup area and that the area is set up as it was when the patches were installed. For example, if you were using /var/adm/patch/backup as a mount point for another file system, make sure that file system is mounted. Once you have solved the /var/adm/patch/backup access or content problem, remove patches through the dupatch Delete Menu.

Output:

```
    Checking patch dependency...
       (depending upon the number of patches you select, this may take a while)
     ---------------------------------------------------------------------------

      - DIGITAL_UNIX_V4.0F / Commands, Shells, & Utility Patches:
            Patch 0019.00 - quota Command Correction

      cannot be deleted.

      Can not find the backup copy for this patch in /var/adm/patch/backup.

     ---------------------------------------------------------------------------
        * Following patch(es) failed in dependency check:

      - DIGITAL_UNIX_V4.0F / Commands, Shells, & Utility Patches:
            Patch 0019.00 - quota Command Correction
```

B.2.2 Patch Removal Blocked by Dependencies on Other Patches

Source: dupatch deletion.

Problem: A specific patch cannot be removed because of its dependency on other installed patches.

Causes: Generally this occurs when you miss the selection of all dependent patches.

Action: Through the dupatch Delete Menu, reselect the patches including the noted dependent patch and try to remove them. The program will notify you of any other dependent patches you might have missed.

Output:

```
Checking patch dependency...
   (depending upon the number of patches you select, this may take a while)
 ------------------------------------------------------------------------

  - DIGITAL_UNIX_V4.0F / Library Patches:
         Patch 0262.00 - libm Corrections

 can not be deleted unless the following patches are also selected or
 deleted first:

  - DIGITAL_UNIX_V4.0F / Library Patches:
         Patch 0676.00 - libm Corrections

 ------------------------------------------------------------------------

    * Following patch(es) failed in dependency check:

  - DIGITAL_UNIX_V4.0F / Library Patches:
        Patch 0262.00 - libm Corrections
```

B.2.3 No Original Files Restored When Patch Is Removed

Source: dupatch deletion.

Problem: The removal of a specific patch results in no original system files being restored.

Causes: This occurs when a patch delivers files to your system that were not shipped in the initial release of the product. For example, the sample output shows the removal of Tru64 UNIX 4.0F Patch 314.00; the patch delivers files that were not shipped with the initial release of Tru64 UNIX 4.0F.

Action: This is an informational message and no further action is required.

Output:

```
=== Deleting "DIGITAL UNIX V4.0F":

     Deleting "Patch: AdvFS Command Correction " (OSFPAT00031400425).
     ------------------------------------------------------------------------

     Patch OSFPAT00031400425 delivered all new files to your system
     so there are no original files to be restored.
     No user action is necessary.

     ------------------------------------------------------------------------
```

B.3 TruCluster Specific dupatch Messages

The following sections show the output of informational messages you might see when running dupatch on a TruCluster system:

B.3.1 System Not Adequately Prepared

Output:

```
This system is part of a V5.0 cluster which has
not been prepared to do a rolling patch installation. Refer to the Patch
Installation Guide as to the proper procedure to start a
rolling patch.
```

B.3.2 Rolling Upgrade in Progress (Installation)

Output:

```
This system is part of a V5.0 cluster which
is currently in the process of being installed via the rolling upgrade/
rolling patch procedure. New patches cannot be installed on the system
until the rolling installation procedure has completed on all cluster
members.
```

B.3.3 Rolling Upgrade in Progress (Baselining)

Output:

```
This Cluster is in the process of a roll. Baselining is not permitted
until the cluster is out of the roll.
```

B.3.4 Version 5.0 Wave 4 Cluster is Unsupported

Output:

```
This system is a Version 5.0 - Wave 4 Cluster. Dupatch cannot patch
this type of cluster. This is an unsupported operation and dupatch will
now exit.
```

B.3.5 Patch Removal Fails Because Needed File Is Unavailable

Source: dupatch deletion.

Problem: An attempt to remove patches fails because the file /var/adm/patch/versionswitch.txt is not available to dupatch.

Cause: At least one of the patches selected for deletion in dupatch has a version switch associated with it (defined by having the attribute PATCH_REQUIRES_VERSION_SWITCH set to "Y" in its *patch.ctrl* file). The versionswitch.txt file is necessary to determine whether the version switch has been thrown.

Action: The dupatch utility returns to the main menu. In order to proceed with the delete operation, you need to determine if the version switch was updated. If it has been thrown, you must run the undo script included with the patch to enable patch deletion (see Section 4.4). If the switch has not been thrown, you can enable the deletion of this patch by reconstructing the versionswitch.txt file. You can also reselect patches for deletion, omitting the patch containing the version switch.

Contact your Customer Service Representative for assistance.

Output:

```
/var/adm/patch/versionswitch.txt file not found!
Cannot delete patches selected since patch_ID requires a version switch.

Please reselect patches or resolve missing /var/adm/patch/versionswitch.txt
Please contact your Customer Service Representative for assistance.
```

B.3.6 Patch Removal Fails Because of a Version Switch

Source: dupatch deletion.

Problem: The deletion of a patch containing a version switch has been blocked because the switch has been thrown.

Action: The dupatch utility returns to the main menu. In order to proceed with the delete operation, you need to determine if the version switch was indeed updated. You can also reselect patches for deletion, omitting the patch containing the version switch.

Output:

```
Version switch thrown for patch patch_ID
You cannot delete patch patch_ID
Please refer to the Patch Kit Release Notes for
instructions on allowing the patch deletion to proceed.
```

B.3.7 dupatch Cannot Create Needed File

Source: Patch installation

Problem: The `dupatch` utility cannot create the file `/var/adm/patch/versionswitch.txt` because it cannot obtain the version switch state from `/etc/sysconfigtab`.

Cause: At least one of the patches selected for installation contains a version switch. `dupatch` records the current version switch state in the file `/var/adm/patch/versionswitch.txt`. In order to facilitate the installation of this patch, this file must be created. While attempting to create this file, `dupatch` could not read the `/etc/sysconfigtab` file

Action: Verify that the file `/etc/sysconfigtab` contains the entry `new_vers_low`.

Output:

```
Cannot obtain version switch info from system files!
Cannot create versionswitch.txt file
Please contact your Customer Service Representative for assistance.
```

B.3.8 Insufficient Free Space (File System Full)

Source: `clu_upgrade setup` stage of the rolling upgrade procedure.

Problem: The rolling upgrade cannot proceed because required space allocations are not met.

Causes: The root (`/`), `/usr`, `/var`, and/or `/i18n` file systems do not have the required amount of free space.

Action: Run the `clu_upgrade -undo setup` command, free up enough space in the affected file systems to meet the requirements listed in Section 4.8.1, and rerun the `clu_upgrade -undo setup` command.

Output:

```
*** Error ***
The tar commands used to create tagged files in the '/' file system have
reported the following errors and warnings:
NOTE: CFS: File system full: /

        tar: sbin/lsm.d/raid5/volsd : No space left on device
        tar: sbin/lsm.d/raid5/volume : No space left on device
```

C

Using dupatch from the Command Line

The dupatch utility provides a command-line interface that allows dupatch to be called by other programs. You can use the command line to invoke all functions except for baselining. The functions have the same operation and definition as the menu-driven interface. For information about using the command-line interface, see the dupatch(8) reference page, which is installed on your system when you install the patch-kit tools and is documented in this appendix.

You must specify all mandatory options on the command line or in a data file. If any mandatory option is missing, the command will fail with an appropriate error message; it will not prompt you for the missing option and information.

C.1 Installing and Removing Release Patch Kits

The following example shows the use of the dupatch command and several of its options to install the Version 5.1B-3 patch kit:

```
/usr/sbin/dupatch -install -kit - license /patch/pk5/patch_kit -name Betty -note \
"installing pk5" -product all -patch all
```

When installing a new kit, the first time you invoke the dupatch command with the -installor -install -precheck_only options you will install the latest patch tools.

The following example shows the use of the dupatch command to remove the Version 5.1B-3 patch kit:

```
/usr/sbin/dupatch -delete -name Joe -note "removing pk" \
-product all -patch T64V51BB26AS0005-20050211
```

The new patch tools cannot be loaded using the delete command on the command line. Doing that will cause the following error to be displayed:

```
product_map does not exist or is empty, Cannot continue.
```

To install the new tools, first issue the install command with the -precheck_only option. This will load the tools and not cause changes to your system. You can then use the delete command.

C.2 Deleting a CSP

You delete a CSP the same way you delete a release patch. The patch number you specify with the -patch option is the CSP's PatchID. The following steps describe how to find the PatchID:

1. Determine which kit the patch was delivered in. For example, T64KIT0020665-V51BB22-ES-20031113.tar. (For information about identifying the fields in this CSP name, see the Patch Kit Overview and Naming document at **http://h30097.www3.hp.com/docs/patch/naming/TITLE.HTM**.)
2. View the text file that ships with kit. For example:

   # **more /var/adm/patch/doc/T64KIT0020665-V51BB22-ES-20031113.txt**

   The text files for CSP kits are also available on the Web at the patch kit download site, **http://www.itrc.hp.com/service/patch/mainPage.do**.

3. Find the section of the text file that lists the PatchID. For example:

```
3 Summary of CSPatches contained in this kit

Tru64 UNIX V5.1B

PatchId   Summary Of Fix
----------------------------------------
C386.00   Fix for SSRT3653, BIND v8
```

4. Type the `dupatch` command line, using the CSP patch ID. For example:

```
# /usr/sbin/dupatch -delete -name "Sally G" -note \
"delete CSP" -product TRU64_UNIX_V5.0B -patch C386.00
```

Notes

- Although a CSP kit can contain multiple patches, not all of them may be installed on your system.
- When deleting a CSP patch, also delete any patches that are required by the patch.

C.3 dupatch Reference Page

This section provides the dupatch reference page, which is installed on your system when you install the patch installation tools.

NAME

dupatch - Installs, deletes and maintains software patch updates to the Tru64 UNIX operating system, the TruCluster software products, and (in later kits) the Worldwide Language Support (WLS) subset.

SYNOPSIS

/usr/sbin/dupatch

/usr/sbin/dupatch -help [-data_file] [-kit *kit_location*] [-patch_id] [-rev] [-product_id]

/usr/sbin/dupatch -install -kit *kit_location* -license -name *user_name* -note *user_note* -patch all | *patch_id* [*patch_id*]... [-cfgfile *config_file*] [-data *data_file*] [-noauto] [-nobackup] [-nolog] [-noroll] [-precheck_only] [-proceed] [-root *root_path*] -product [all | *product_id*] [-single_user]

/usr/sbin/dupatch -delete -name *user_name* -note *user_note* -patch all | *patch_id* [*patch_id*]... [-cfgfile *config_file*] [-data *data_file*] [-noauto] [-nolog] [-noroll] [-proceed] [-root *root_path*] [-product all | *product_id*] [-single_user]

/usr/sbin/dupatch -track -type [patch_level | file | kit | patch] [-data *data_file*] [-kit *kit_location*] [-nolog] [-root *root_path*]

COMMAND KEYWORDS

`-install` *`install-options`*

Installs a software patch or patch kit.

`-delete` *`delete-options`*

Removes an installed patch or patches from the operating system. Patch deletion requires that the patch was installed as a reversible patch.

`-track` *`track-options`*

Constructs a history of patch installations and deletions. Information can be patch-kit specific or patch-file specific.

`-help` *`help-options`*

Requests quick help on dupatch. Supplying an argument will provide help specifically on that argument.

OPTIONS

Required –install Options

`-kit` *`kit_location`*

Specifies the location of the patch kit from which patches will be installed onto the system.

`kit_location` is a full path to the directory containing the patch kit.

`-license`

Specifies that you have read and agreed to the license required to install the patch kit. This option is required for Version 5.1B-3 and higher. If you do not specifiy this option when required, you will see the following message:

"Please read the license agreement (license.txt) in the top level directory of the patch kit. To accept the license agreement, include the -license option in the command line."

You can also read the license in the *Patch Summary and Release Notes* document that is included with your kit.

When you specify this option, the following message is displayed:

"You have accepted the license agreement."

`-name` *user_name*

Specifies the name to be recorded in `event.log`. Enclose the *user_name* in quotation marks if it contains space characters.

`-note` *user_note*

Records user-supplied text in the event log. The *user_note* is a text string enclosed in quotation marks.

`-product all |` *product_id [product_id]...*

Required when more than one product is installed.

Specifies the installed operating system and TruCluster software when installing patches from an old style patch kit. Product ID specifications are not case sensitive. Wildcard characters are not permitted.

When installing an inclusive patch kit, the use of `all` is mandatory. See Specifying a Product ID with –product.

`-patch all |` *patch_id [patch_id]...*

Directs `dupatch` to install all (`all`) patches or specific (*patch_id*) patches from the specified patch kit.

When installing an inclusive patch kit, the use of `all` is mandatory. See Specifying a Patch ID with –patch.

Optional –install Options

`-cfgfile` *config_file*

Specifies a configuration file for rebuilding the kernel. See Specifying a Configuration File.

`-data` *data_file*

Specifies a file that contains arguments (in the form *argument = value*) to the `dupatch` command. See Using a Data File.

`-noauto`

Directs `dupatch` to not automatically rebuild the kernel if indicated by the patches installed. In addition, if running `dupatch` to install the patches in

single-user mode, the system will not automatically reboot after the patch process is complete.

`-nobackup`

Directs `dupatch` to not retain backup information during a patch installation. This will remove the ability to back out an installed patch.

`-nolog`

Directs `dupatch` to not record actions in a `session.log` file.

`-noroll`

Directs `dupatch` to install patches on a cluster using the no-roll procedure rather than the default rolling-upgrade procedure.

`-precheck_only`

Directs `dupatch` to perform the preinstallation check but to not proceed with the patch installation. If `-precheck_only` is omitted, `dupatch` begins the installation process after the preinstallation check has been completed, as long as no patch failed the preinstallation check. The preinstallation check determines whether new patches that depend on the presence of other patches or software subsets can be installed. It does this by verifying that the required patches or software subsets are already installed onto the system.

`-proceed`

Directs `dupatch` to install any patches that passed the preinstallation check, even if one or more patches failed the preinstallation check. If `-proceed` is omitted, `dupatch` will not install any patches if at least one patch fails the preinstallation check. The preinstallation check determines whether new patches that depend on the presence of other patches or software subsets can be installed. It does this by verifying that the required patches or software subsets are already installed onto the system.

`-root` *`root_path`*

Specifies an alternate root location. The default *`root_path`* is `/` for all operations.

`-single_user`

If the system is presently in multiuser mode, brings the system down to single-user mode prior to installing patches.

`-rev`

Prints the current dupatch revision.

Required –delete Options

`-name` *`user_name`*

Specifies the name to be recorded in `event.log`. Enclose the *`user_name`* in quotation marks if it contains space characters.

`-note` *`user_note`*

Records user-supplied text in the event log. The *`user_note`* is a text string enclosed in quotation marks.

`-product all|`*`product_id [product_id]...`*

Mandatory when more than one product is installed.

Specifies the installed operating system and TruCluster software when removing patches from an old sytle patch kit. Product ID specifications are not case sensitive. Wildcards are not permitted.

When removing an inclusive patch kit, the use of `all` is mandatory. See Specifying a Product ID with –product.

`-patch all |` *`patch_id [patch_id]...`*

Directs `dupatch` to remove all (`all`) patches or specific (*`patch_id`*) patches from the specified patch kit.

When removing an inclusive patch kit, the use of `all` is mandatory. See Specifying a Patch ID with –patch.

Optional –delete Options

`-data` *`data_file`*

Specifies a file that contains arguments (in the form *`argument = value`*) to the `dupatch` command. See Using a Data File.

`-nolog`

Directs `dupatch` to not record actions in a `session.log` file.

`-noroll`

Directs `dupatch` to remove patches on a cluster using the no-roll procedure rather than the default rolling-upgrade procedure.

`-proceed`

Directs `dupatch` to delete any patches that passed the predeletion check, even if one or more patches failed the predeletion check. If `-proceed` is omitted, `dupatch` will not delete any patches if at least one patch failed the predeletion check. The predeletion check determines whether any installed patches have dependencies on any of the patches listed for removal. If such dependencies exist, `dupatch` blocks the removal of any required patch.

`-root` *`root_path`*

Specifies an alternate root location. The default *`root_path`* is `/` for all operations.

Required –track Options

-type *patch_level*
-type *file*
-type *kit*
-type *patch*

Provides a single command (*patch_level*) that lists a full description of the patch kits, CSPs, and ERPs installed on your system, or lists all patched files (-file), installed patch kits (-kit), or installed patches (-patch).

Optional –track Options

-data *data_file*

Specifies a file that contains arguments (in the form *argument = value*) to the dupatch command. See Using a Data File.

-kit *kit_location*

Identifies the location of the patch kit for which the reports will cover.

kit_location is a full path to the directory containing the patch kit.

-nolog

Directs dupatch to not record actions in a session.log file.

-root *root_path*

Specifies an alternate root location. The default *root_path* is / for all operations.

DESCRIPTION

The dupatch utility is an interactive program used to install and delete software patches to the Tru64 UNIX operating system and systems running TruCluster software products.

With dupatch you can baseline your system to incorporate any system files that may have been manually installed. You can also use dupatch to obtain a list of installed patches or view the system history of patch installations and deletions.

When invoked without arguments, dupatch is run interactively by providing menus that step you through the patching procedure while prompting you for necessary information. Alternatively, you can invoke dupatch from the command line, whereby you supply required arguments to the dupatch command.

Although you can install patches in either single-user or multiuser mode, the use of single-user mode is strongly recommended. In multiuser mode, libraries and system files that are in use by active processes may be affected by the new patches. The patching of any active library or system files may result in unexpected consequences.

Beginning with Version 5.1B Patch Kit 4 (base level 25), patch kits are packaged as “inclusive patch kits,” which require all patches in the kit to be installed or removed together. Therefore, you cannot use the following options with an inclusive patch kit:

- /usr/sbin/dupatch -install -patch *patch_id*
- /usr/sbin/dupatch -delete -patch *patch_id*

Attempting to use the *patch_id* option will cause the command to fail.

Inclusive patch kits will also install patches for the Worldwide Language Support (WLS) subset if the WLS subset is installed on your system.

On clustered systems running TruCluster software Version 5.0A or higher, the dupatch utility is run in conjunction with the rolling upgrade procedure. (See the *Patch Kit Installation Instructions* or the *Cluster Installation* manual for information about performing a rolling upgrade.)

Using a Data File

The *data_file* that you specify with the -data option is a fully qualified file location and a file that contains command-line options with the following format:

```
option1 = value
option2 = value
  .
  .
option3 = n
```

For example:

```
kit  = /mnt
name  =  Joe
note  =  Installing April patch kit
product = Tru64_UNIX_V5.1
patch   = 27.01 63.00 74 83.01
product = TruCluster_V5.1
# multiple patches are separated by space characters
patch   = 21.01 27.01 40
precheck_only
nobackup
```

Blank lines and comments (preceded with #) are allowed. Line continuation (\ ) is required if a specification spans multiple lines. Only one *data_file* is permitted per command line and nested *data_file* specifications are not allowed.

Specifying a Product ID with –product

When installing or removing an inclusive patch kit, you must specify all with the -product option. For example:

```
./dupatch -install -product all -patch all -name Joe -note \
"installing pk4" -kit .
```

For old style patch kits, the *product_id* you specify with -product is one of the following:

```
TRU64_UNIX_V5.1B
TRU64_UNIX_V5.1A
TRU64_UNIX_V5.1
TRU64_UNIX_V5.0A
TRU64_UNIX_V5.0
TRU64_UNIX_V4.0G
TRU64__UNIX_V4.0F
DIGITAL_UNIX_V4.0D


TruCluster_V5.1B
TruCluster_V5.1A
TruCluster_V5.1
TruCluster_V5.0A
TruCluster_V1.6
TruCluster_V1.5
```

- A *product_id* specification is not necessary when the system being patched has only one product installed; for example, Tru64 UNIX Version 4.0F with no TruCluster software product.
- A *product_id* specification only applies to the *patch_id* specifications that follow it and ends when another *product_id* is specified.
- Because the purpose of the *product_id* is to clarify the *patch_id* specification, the *product_id* must precede the *patch_id*.
- Product strings are not case sensitive. Wildcard characters are not permitted.

The following example shows the use of a product string with an old style patch kit:

```
/pk3/patch_kit/dupatch -install -product DIGITAL_UNIX_V4.0F -patch 1.1 \
-product TruCluster_V1.6 -patch 35 -name Joe -note \
"installing patch 1.1" -kit /pk3/patch_kit
```

Specifying a Patch ID with –patch

You must specify `all` with the `-patch` option when installing or removing an inclusive patch kit. For example:

```
./dupatch -install -product all -patch all \
-name Joe -note "installing pk4" -kit .
```

For old style patch kits, the *patch_id* you specify with the `-patch` option has the following format:

```
xxxx[.yy]
```

For example:

```
15
200.11
10.2
00111.02
```

- Both xxxx and yy are numeric values; leading zeros can be omitted.
- Patch revision (yy), when left unspecified, maps to wildcarded "??"
- Multiple *patch_id* specifications are separated by white space.
- The keyword `all` cannot be combined with other patch IDs.
- If *product_id* is used, *patch_id* must come after it.

The following example shows the use of the `-patch` option with an old style patch kit:

```
/pk3/patch_kit/dupatch -install -product DIGITAL_UNIX_V4.0F -patch 1.1 \
-product TruCluster_V1.6 -patch 35 -name Joe -note \
"installing patch 1.1" -kit /pk3/patch_kit
```

Specifying a Root Path

The *root_path* you specify with the `-root` option specifies an alternative root for the specified operation. (The `-root` option is similar to the `-D` option of `setld`.) The following list describes characteristics of the `-root` option.

- The root path must be the root of a complete UFS file system or AdvFS domain.
- The default root path is `/` for all operations.
- If `-root` is the only argument on the command line, `dupatch` will proceed in interactive mode; this is an exception to the command-line rule previously mentioned.
- When performing an alternate root installation, the `-noauto` flag is set implicitly.

Specifying a Configuration File

The -cfgfile option to the -install and -delete command options allows you to call in the system configuration file (/usr/sys/conf/*config_file*). For information about creating or modifying a config_file, see the doconfig(8) and sizer(8) reference pages.

RESTRICTIONS

The following restrictions apply to the dupatch utility.

You must be logged in as root to run dupatch.

The system must be running in single-user mode when removing patches.

The -product option must precede the -patch option on the command line.

EXIT STATUS

0 (Zero)	Success.
>0	An error occurred.

ERRORS

See the Patch Kit Installation Instructions for a detailed list of dupatch error messages.

EXAMPLES

1. The following interactive example shows how to invoke the menu-driven interface of dupatch:

```
# dupatch

Tru64 UNIX Patch Utility (Rev. 46-00)
=========================
 - This dupatch session is logged in /var/adm/patch/log/session.log

    Main Menu:
    ---------

    1)  Patch Installation
    2)  Patch Deletion

    3)  Patch Documentation
    4)  Patch Tracking

    5)  Patch Baseline Analysis/Adjustment

    h)  Help on Command Line Interface

    q)  Quit

Enter your choice: 1
```

2. The following interactive example shows how to perform a preinstallation check on patch 00183.00 contained in the kit located at */mnt/patch_kit*. This will verify that the specified patch can be installed onto the system without actually proceeding with the installation:

```
# dupatch -install -kit /mnt/patch_kit  -name Jessica -note \
"Pre-Installation check only on 183.00" -patch 183.00 -precheck_only
```

3. The following interactive example shows how to install all patches in kit located at */mnt/patch_kit*:

```
# dupatch -install -kit /mnt/patch_kit  -name Jessica \
-note "install all patches" -patch all
```

4. The following interactive example shows how to identify all patches installed on system:

```
# dupatch -track -type patch
```

5. The following interactive example shows how to list all system files updated by installed patches:

```
# dupatch -track -type file
```

6. The following interactive example shows how to remove patch 00183.00 from the system. Note that the system will automatically be rebooted upon patch deletion because `-noauto` was not specified:

```
# dupatch -delete -patch 183.00 -name Joe \
-note "delete patch 00183.00 from system"
```

7. The following interactive example shows how to obtain help on specifying *patch_id* usage:

```
# dupatch -help patch_id
```

ENVIRONMENT VARIABLES

The following environment variables affect the execution of `dupatch`:

`MAX_LOGS`

Specifies the maximum number of session logs to be retained on the system. The default number is 25. If, for example, `MAX_LOGS` is set to 25, the oldest session log would be named `session.log.24` and the current would be named `session.log`, with no number affixed.

`_ROOT`

Overrides the location of the root directory. The default value is /, the system root directory. This value must be the top-level directory of a file system (or an AdvFS domain).

`PATCHDIR`

Specifies the path to the patch tools repository. The default value is `$_ROOT/var/adm/patch`.

FILES

`/var/adm/patch/log/session.log.`*n*

This file captures `dupatch` activities. A separate session log is written with each `dupatch` session and log files from the previous sessions are saved. The order is first in, first out, with `session.log.`*$MAX_LOGS* as the oldest file.

`/var/adm/patch/log/Dupatch_load_`*Date*`.log`

This file specifies the date when the patch tools were loaded or updated onto the system.

`/var/adm/patch/log/baseline.log.`*n*

This file records the screen output from the baselining session. A separate baseline log is written for each baselining session and log files from previous sessions are saved. The order is first in, first out, with `session.log.`*$MAX_LOGS* as the oldest file.

`/var/adm/patch/log/event.log.`*n*

This file captures information regarding patch installation and removal operations. A separate event log is written each time patches are installed or removed. Log files from previous sessions are saved. The order is first in, first out, with `session.log.`*$MAX_LOGS* as the oldest file.

`/var/adm/patch/backup`

The files in this directory are used to restore the system to its former state if patches are deleted.

`/var/adm/patch/doc/OSFPAT*`*patch_no*`.abs`

Provides brief summary of what a patch fixes.

`/var/adm/patch/doc/OSFPAT*`*patch_no*`.txt`

Provides detailed discussion of what a patch fixes.

root-path`/usr/.smdb./OSFPAT*.inv`

Lists the subset inventory files.

root-path`/usr/.smdb./OSFPAT*.ctrl`

Lists the subset control files.

root-path`/usr/.smdb./OSFPAT*.scp`

Lists the subset inventory programs.

root-path`/usr/.smdb./OSFPAT*.lk`

Lists the subset installed lock files.

SEE ALSO

Commands: `setld`(8), `clu_upgrade`(8)

Documents:

Patch Kit Installation Instructions

Patch Summary and Release Notes for the patch kit to be installed

Tru64 UNIX *Installation Guide*

Tru64 UNIX *System Administration* guide

TruCluster Software Products *Software Installation* guide

TruCluster Software Products *Cluster Administration* guide

D

Inclusive Patch Kits

Beginning with Version 5.1B-2, we changed to the way Tru64 UNIX patch kits are installed. If you did not install Version 5.1B-2 but have installed earlier kits, you will see the following differences in the kitting process:

- All or none installation

  When you install an inclusive patch kit, you must install all patches; you can no longer select specific patches to install. By making the installation of all patches mandatory, you can patch with greater confidence that the process will be problem free.

  Before a patch kit is released, it is tested on many types of systems and system configurations. This testing continues until we are assured that the patches perform the tasks they were designed for and do not introduce new problems. It is not possible to achieve this type of testing on every possible combination of individually selected patches.

- Substantially reduced installation time

  The installation process for inclusive patch kits can reduce the time it takes to install the patches by as much as half from what you are used to. For large, clustered systems, the difference can be several hours faster.

- Fewer patches displayed

  Because of the way these new patch kits are designed, you will see many fewer patches listed by `dupatch` during the installation process. For example, a partial listing you see will be similar to the following:

```
- Tru64_UNIX_V5.1B / Security Related Patches:
      * Patch 26001.00 - SP04 OSFACCT540

      * Patch 26002.00 - SP04 OSFADVFS540 (SSRT2275)

      * Patch 26003.00 - SP04 OSFADVFSBIN540
```

  In the old-style patch kits, these three patches might have consisted of perhaps 20 individual patches being displayed. The difference is not in the content of the kits, but rather in the way the patches are packaged and installed. In this example, the `SP04` identifies the patch as belonging to Version 5.1B-2, the `OSF...540` identifies the subset the patch is included in, and the `SSRT2275` indicates a type of security patch.

  As with previous kits, you can find a brief overview of all the patches (listed by patch number) in the kit's *Patch Summary and Release Notes*.

- All or none patch removal

  As with the installation process, if you want to remove a patch, you must remove all of them. That is, you can no longer select individual patches for removal.

- Patches for Worldwide Language Support (WLS) subset

  The inclusive patch kits deliver patches that may be required for the WLS subset. As with the TruCluster Server patches, the WLS patches will only be installed if you have the WLS subset installed.

Glossary

baselining

A dupatch feature that looks at the files installed on a system, compares them to the files it expects to find, and prevents the installation of any patch files that might cause an incompatibility among system files.

Customer-Specific Patch (CSP) Kit

A patch kit that is developed and made available to resolve a problem for a specific customer. A Customer-Specific patch is developed with prior knowledge of that customer's unique hardware and software configuration and environment. Customer-Specific patches may not be useful for another customer's system. An Early Release patch is a type of CSP.

See also *Early Release Patch (ERP) Kit*, *Release Patch Kit*

dupatch

A utility included in a patch kit that installs, removes, and manages patches for Tru64 UNIX and TruCluster software products. This utility is installed and left on the system through the successful installation of a patch kit.

Early Release Patch (ERP) Kit

A patch kit that contains a patch or patches that will be included in a Release Patch Kit that is still under development. ERPs, which are a type of Customer-Specific patch, are provided by HP to help customers who have an immediate need for some specific functionality that will be included in an upcoming Release Patch Kit.

See also *Customer-Specific Patch (CSP) Kit*, *Release Patch Kit*

force install

A term somtimes used to describe the abilility of the baselining procedure to enable the installation of patches that are blocked by the intallation procedure.

inclusive patch kit

See *new style patch kit*

new style patch kit

Also called an inclusive patch kit, a new style patch kit is a Release Patch Kit that provides an improved way of delivering patches. Among the ways that a new style patch kit differs from its predecessors is that it requires an all or none installation and removal of the patches in that kit. The first Tru64 UNIX new style patch kit was Version 5.1B Patch Kit 4 (Base Level 25).

See also *Release Patch Kit*

no-roll patching

A process that patches your cluster in one operation and requires only one reboot of the whole cluster to complete the operation. This method was developed for mission-critical environments to provide a way to apply patches quickly, with a minimum amount of down time.

The no-roll patch process is a modification of dupatch; that is, all patches are installed or removed entirely using the dupatch utility, as opposed to the clu_upgrade and dupatch utilities used in the rolling upgrade procedure. The no-roll process conducts significantly fewer operations than the rolling upgrade procedure.

See also *rolling upgrade*

official patch

See *Release Patch Kit*

old style patch kit

See *new style patch kit*

patch

A file or a collection of files that contain fixes to problems. When possible, patches are merged together into one patch if they have intersecting files or codependencies. A patch may correct one or more problems.

Each patch is packaged in its own `setld` subset. The subsets are managed by a utility named `dupatch`.

patch applicability

A file-by-file check of system files to determine whether a patch might cause a system to be degraded or crash. The installation of a patch is blocked if any system files to be replaced by that patch are not valid predecessors of the patch files.

Release Patch Kit

A patch kit that HP provides to modify a specific version of the Tru64 UNIX operating system and TruCluster software. Sometimes referred to as official patch kits, Release Patches Kits are intended for worldwide distribution and can be safely used on any customer's system within the guidelines documented in the kit. The patches in a Release Patch Kit are referred to as Release patches.

See also *Customer-Specific Patch (CSP) Kit*, *Early Release Patch (ERP) Kit*, *new style patch kit*

rolling upgrade

A software upgrade of a cluster that is performed while the cluster is in operation. One member at a time is rolled and returned to operation while the cluster transparently maintains a mixed-version environment for the base operating system, cluster, and Worldwide Language Support (WLS) software. Clients accessing services are not aware that a rolling upgrade is in progress.

On Version 5.0A and higher systems, you use a rolling upgrade to patch a cluster or to update the Tru64 UNIX operating system or TruCluster software on a cluster. The procedure is the same for both types of upgrades — the only difference is the command you run during the install stage of the rolling upgrade procedure.

See also *no-roll patching*

setld

An interactive program for installing and managing software subsets. Software products are organized into subsets that can be loaded, deleted, inventoried, and configured. The load operation reads software from disk, tape, CD-ROM, or an Internet installation server. The patch installation tool, `dupatch`, is based on the `setld` program.

tar file

A file created with the `tar` command that saves and restores multiple files in a single file. Tru64 UNIX patch kits are provided as tar files (except for kits included on a Patch CD–ROM).

version switch

During a rolling upgrade, a version switch manages the transition of the active version to the new version of an operating system. The active version is the one that is currently in use. The purpose of a version switch in a cluster is to prevent the introduction of potentially incompatible new features until all members have been updated.

See also *rolling upgrade*

Index

A

applicability of patches, 1-3
applications
(*See* layered products)

B

backup
default location for installed patches, 2-2
during a rolling upgrade, 4-16n
performing before baselining, 2-4
relation to patch reversibility, 1-3
backup directory
using as mount point for separate disk partition, 2-2
baseline.log, A-1
baselining
handling manually installed system files with, 2-3
reporting information on layered products, 2-5, 2-8
warnings when using, 2-4n
bcheckrc command
single-user mode on single system, 3-2
single-user mode with cluster, 4-7
use to roll cluster member, 4-8
blocking layered products, 4-25

C

CAA
testing lead member during a rolling upgrade, 4-8
caution
blocking layered products and rolling upgrade, 4-23n
creating tagged files, 4-18n
not deleting files in /var/adm/update, 4-18n
cfsmgr command, 4-7
clean stage, 4-21t
clu_upgrade command
check setup command, 4-5, 4-16t
clean command, 4-10, 4-21t
commands for, 4-11
completed command, 4-11
log file, 4-18
log file for, 4-22
postinstall command, 4-8, 4-20t
preinstall command, 4-6, 4-19t
roll command, 4-8, 4-20t
setup command, 4-6, 4-18t
switch command, 4-10, 4-21t
tagged add command, 4-24
tagged check command, 4-23
tagged disable command, 4-24
tagged enable command, 4-24
undo command, 4-12
version command, 4-13
cluamgr command, 4-8
cluster alias
RIS registration, 4-26
testing lead member during a rolling upgrade, 4-8
cluster creation
patching prior to, 1-7
clusterwide file systems
free space required for a rolling upgrade, 4-17
command line
installing and removing patches from, C-1
restriction on loading new dupatch tools from, C-1
common installation steps, 3-4
CSP
deleting using command line, C-1
dependence on patch kit, 3-8
file found when baselining, 2-4
information about when patch tracking, 1-4
installing and removing, 3-1
installing with dupatch, 1-1
patch applicability of, 1-3
patch installation blocked by, B-5
problem during preinstallation check, 3-5
removing, 3-8
reversibility of, 1-3
superseded in Release kits, 1-8
Ctrl/c
restriction on using during patch installation, 1-7
Customer-Specific patch kit
(*See* CSP)
customized files
message in session log, 1-7

D

Dataless Management Services
(*See* DMS)
DEC_VERSION_TAG property, 4-22
default cluster alias
testing lead member during a rolling upgrade, 4-8
deleting patches
(*See* removing patches)
disk space
requirement for a rolling upgrade, 4-17
where to find required amount for patch installation, 3-1
DMS
restriction using, 1-7
documentation
in files used by dupatch, 1-4
dupatch
calling from other programs, C-1
command-line interface, C-1
described, 1-1
in rolling upgrade install stage, 4-19t
loading new patch tools with, 1-2
log files created by, A-1
menu determined by login location, 2-1
reference page for, C-2
restriction on loading new dupatch tools from command line, C-1
tracking information with, 1-4
viewing patch documentation with, 1-4
dupatch tools
installing from command line, C-1
restriction on loading from command line using delete, C-1
Dupatch_load_date.log, A-1

E

Early Release Patch Kit
(*See* ERP)
ERP, 3-1
(*See also* CSP)
information about when patch tracking, 1-4
installing and removing, 3-1
installing with dupatch, 1-1
removing, 3-8
error messages, B-1
event.log, A-1

F

force install, Glossary-1

H

halting an installation, 1-7

I

i18n
(*See* WLS)
init command
single-user mode on single system, 3-2
single-user mode with cluster, 4-7
use to roll cluster member, 4-8
install stage, 4-19t
installing successive patch kits, 3-2
installupdate command, 4-7, 4-19t

K

kernel
rebuilding, 3-5
kloadsrv command
single-user mode on single system, 3-2

L

layered products
blocking, 4-25
error caused by product collision, B-2
general guidelines for rolling upgrade, 4-25
potential problems when upgrading system, 1-8
report on when baselining, 2-5, 2-8
lead member, 4-16, 4-23
lmf reset command
single-user mode on single system, 3-2
single-user mode with cluster, 4-7
use to roll cluster member, 4-8
log files, A-1
baseline.log, A-1
clu_upgrade command, 4-18
Dupatch_load_date.log, A-1
event.log, A-1
untar.log, 3-2

M

member boot disk
rolling upgrade space requirement, 4-17
menu
(*See* dupatch)
messages
(*See* error messages)
multiple (parallel) rolls, 4-9
multiuser mode
restriction on patching from, 1-6

N

Network File System
(*See* NFS)

New Hardware Delivery kits
(*See* NHD)
NFS
support for patch installation, 1-7
NHD
installing kits during a rolling upgrade, 4-2
nhd_install command, 4-7, 4-19t
no-roll patching, 5-1
(*See also* rolling upgrade)
overview, 5-1
steps for performing, 5-2

O

.Old.. files, 4-22

P

parallel rolls, 4-9
patch kit directory
removing after patch installation, 3-8
patch tools
available on the Web
(*See* dupatch)
patching prior to cluster creation, 1-7
postinstall stage, 4-20t
preinstall stage, 4-19t
preinstallation
checklist, 3-1
creating a baseline, 2-3
preinstallation check
example of patch that fails, 2-1
performing, 2-1
preparation stage, 4-16t
procedure for using dupatch, 3-11

R

rebooting system
in multiuser mode, 3-7
in single-user mode, 3-7
reference page for dupatch, C-2
Remote Installation Services
(*See* RIS)
removing patches
during a rolling upgrade, 4-10
example of from command line, C-1
overview, 3-8
restriction on using ALL of the above menu item, 3-9
that contain customized files, 1-7
using dupatch deletion option, 3-11
removing temporary patch kit directory, 3-8
reversibility of patches
(*See* reverting systems to prior state)
reverting systems to prior state, 1-3
RIS
registering the default cluster alias, 4-26
restriction using, 1-7
roll stage, 4-20t
parallel rolls, 4-9
rolling patch
(*See* rolling upgrade)
rolling upgrade, 4-1
(*See also* no-roll patching)
backups during, 4-16n
checking the status of, 4-11
disk space requirements, 4-17
lead member, 4-16, 4-23
NHD, 4-2
parallel rolls, 4-9
patch, 4-2
procedure, 4-5
stages of
(*See* rolling upgrade stages)
tagged files, 4-22
testing the lead member with CAA, 4-8
testing the lead member with cluster alias, 4-8
undoing a stage, 4-12
unsupported tasks, 4-4
updating the Tru64 UNIX operating system disk, 4-3n
version switch, 4-24
rolling upgrade stages
clean, 4-21t
install, 4-19t
postinstall, 4-20t
preinstall, 4-19t
preparation, 4-16t
roll, 4-20t
setup, 4-18t
switch, 4-21t
rolls_ver_lookup attribute, 4-18, 4-22
root file system
rolling upgrade space requirement, 4-17

S

session log, A-1
message in when removing a patch, 1-7
setld command
restriction when adding or removing patches, 1-7
setup stage, 4-18t
simultaneous (parallel) rolls, 4-9
single-user mode

recommendation on patching from, 1-6
stages
checking the status of, 4-11
parallel rolls, 4-9
time required to perform, 4-5
undoing, 4-12
storage space
(*See* disk space)
switch stage, 4-21t
system upgrades
potential problems with, 1-8

T

tagged files, 4-22
creating during setup stage, 4-18
creating for OSF, TCR, and IOS product codes, 4-23
removing during clean stage, 4-21
rules for creating, 4-23
verifying during preinstall stage, 4-19
verifying existence of, 4-12
tar file
expanding, 3-1
tools
(*See* dupatch tools)
tracking information, 1-4

U

unsupported tasks
during a rolling upgrade, 4-4
untar.log, 3-2
updadmin command, 4-22
/usr file system
rolling upgrade space requirement, 4-17

V

/var file system
rolling upgrade space requirement, 4-17
/var/adm/patch/backup, 1-3, 1-7, 2-2, 3-4, B-4, B-6
version switch, 4-24
enabling new functions with, 3-7
in no-roll patch procedure, 5-3
overview, 1-6
versw command, 4-24
vMAC
algorithm for creating a hardware address, 4-26

W

Web sites
(*See* support)
WLS
installing on patched system, 3-8
rolling upgrade space requirement, 4-17