HITACHI
DATA SYSTEMS

Hitachi HiCommand™ Device Manager Server Installation and Configuration Guide

Trademarks

Hitachi Data Systems and the Hitachi Data Systems design mark are registered trademarks service marks of Hitachi, Ltd.

HiCommand is a trademark of Hitachi Data Systems Corporation.

InterBase, InterServer, and InterClient are trademarks or registered trademarks of Borland Software Corporation.

Unicenter TNG is a registered trademark of Computer Associates International, Inc.

CommandView is a trademark or registered trademark of Hewlett-Packard Company.

Pentium is a registered trademark of Intel Corporation.

Internet Explorer, Windows 95, Windows 98, Windows ME, Windows NT, Windows XP, and Windows 2000 are trademarks or registered trademarks of Microsoft Corporation.

Netscape is a registered trademark of Netscape Communications Corporation.

SPARC is a trademark or registered trademark of SPARC International, Inc. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

Java, Java2, Java Runtime Environment (JRE), Java Web Start, Solaris, and StorEdge are trademarks or registered trademarks of Sun Microsystems, Inc.

VERITAS and SanPoint Control are trademarks or registered trademarks of VERITAS Software Corporation.

UNIX is a registered trademark of X/Open Company Limited in the United States and other countries and is licensed exclusively through X/Open Company Limited.

InstallAnywhere is a trademark or registered trademark of Zero G Software, Inc.

All other brand or product names are or may be registered trademarks, trademarks or service marks of and are used to identify products or services of their respective owners.

Notice of Export Controls

Export of technical data contained in this document may require an export license from the United States government and/or the government of Japan. Contact the Hitachi Data Systems Legal Department for any export compliance questions.

Document Revision Level

Revision	Date	Description
MK-91HC002-0	September 2001	Initial Release
MK-91HC002-1	October 2001	Revision 1, supersedes and replaces revision 0
MK-91HC002-2	November 2001	Revision 2, supersedes and replaces revision 1
MK-91HC002-3	January 2002	Revision 3, supersedes and replaces revision 2
MK-91HC002-4	February 2002	Revision 4, supersedes and replaces revision 3
MK-91HC002-5	June 2002	Revision 5, supersedes and replaces revision 4
MK-91HC002-6	November 2002	Revision 6, supersedes and replaces revision 5
MK-91HC002-7	May 2003	Revision 7, supersedes and replaces revision 6

Software Revision Level

This document is written to support version 2.3 of the HiCommand™ Device Manager server software.

Changes in this Revision

- Added description of HiCommand™ Suite common component (sections 1.3 and 1.4)
- Revised instructions for installing Device Manager for Windows® (section 2.2)
- Added new section on verifying HiCommand™ installation for Windows® (section 2.4)
- Revised instructions for upgrading or reinstalling the Device Manager server for Windows® (section 2.6)
- Revised instructions for uninstalling the Device Manager server for Windows®, and InterBase® Server and Client for Windows® (section 2.7)
- Revised instructions for uninstalling InterClient for Windows® (section 2.8.2)
- Revised instructions for restoring an existing database for Windows® (section 2.8.3)
- Revised instructions for restoring an empty database for Windows® (section 2.8.4)
- Revised instructions for installing Device Manager for Solaris™ (section 3.2)
- Added new section on installing the common component for Solaris™ (section 3.3.3)
- Added new section on verifying installation for Solaris™ (section 3.4)
- Revised instructions for upgrading or reinstalling Device Manager for Solaris™ (section 3.5)
- Revised instructions for uninstalling InterBase® software for Solaris™, and Device Manager for Solaris™ (section 3.6)
- Added new chapter on the HiCommand™ Suite common component (Chapter 4)
- Added new properties server.base.home (section 5.2.1.25), logger.hicommandbase.loglevel (section 5.2.3.3), logger.hicommandbase.sysloglevel (section 5.2.3.4), logger.hicommandbase.MaxBackupIndex (section 5.2.3.5), and logger.hicommandbase.MaxFileSize (section 5.2.3.6); Deleted property logger.errorlogfile (former section 5.2.3.4)
- Added new section on using Trouble Information Acquisition (TIA) to obtain batch error information (section 6.2)
- Added new section on using hcmdsras command to obtain error information for the common web service and single sign-on (section 6.3)

Preface

This document provides instructions for installing the HiCommand™ Device Manager server software. Server installation also includes Java2™ JRE™ version 1.3.1_03 (installation on the server is transparent to the user), InterBase® version 6.0.1 server software, and the InterClient® 2.0 JDBC Driver for InterBase® systems. Please read this manual and the associated HiCommand™ Device Manager documents carefully to understand how to successfully install and configure your HiCommand™ Device Manager server for storage management, access control and system support.

This user's guide assumes that:

- the user has a background in data processing and understands direct-access storage device subsystems and their basic functions,
- the user is familiar with the Hitachi Lightning 9900 V Series, Hitachi Lightning 9900, Hitachi Thunder 9500V, Hitachi Thunder 9200, or Sun® StorEdge™ T3 storage subsystems,
- the user is familiar with the Solaris™, Windows® NT, or Windows® 2000 operating systems.

Please contact your Hitachi Data Systems account team or refer to the Hitachi Data Systems worldwide web site (http://www.hds.com) for additional information on the Lightning 9900™ V Series or Lightning 9900™ subsystem, or the Thunder 9500V or 9200 storage subsystem, and their features and functions.

Note: The use of HiCommand™ Device Manager and all other Hitachi Data Systems products is governed by the terms of your license agreement(s) with Hitachi Data Systems.

Note: The use of the Sun® StorEdge™ T3 Array and all other Sun® products is governed by the terms of your license agreement(s) with Sun Microsystems, Inc.

COMMENTS

Please send us your comments on this document: doc.comments@hds.com.

Make sure to include the document title, number, and revision.
Please refer to specific page(s) and paragraph(s) whenever possible.

(All comments become the property of Hitachi Data Systems Corporation.)

Thank you!

Contents

List of Figures

List of Tables

Chapter 1 Introduction to HiCommand™ Device Manager

1.1 Overview of HiCommand™ Device Manager

HiCommand™ Device Manager provides a consistent, easy to use, and easy to configure interface for managing storage products. As a product of a Hitachi Data Systems and Sun® Microsystems partnership, Device Manager provides a graphical client interface for real-time interaction with managed storage arrays as well as a command line interface (CLI) for scripting. Device Manager gives storage administrators access to the configuration, monitoring, and management features that are already integrated into existing Hitachi Data Systems software products. Device Manager allows you to view the configuration of the storage arrays added to the Device Manager system, and perform configuration operations such as allocating storage or securing LUNs.

HiCommand™ Device Manager provides:

- Storage subsystem discovery and configuration display
- Hierarchical group management for storage
- Alert presentation
- Volume (LUN) configuration
- Management of hosts and WWNs
- Several levels of access and functionality for end users, including Access Control, Storage Management and System Support:
  - Access Control handles support for the system administrator, storage administrator, maintenance user and guest user
  - Storage Management functions include storage configuration and manipulation
  - System support functions include user administration, host agent activity and security

Note: The use of the HiCommand™ Device Manager product and all Hitachi Data Systems products is governed by the terms of your license agreement(s) with Hitachi Data Systems and/or Sun® Microsystems.

1.2 HiCommand™ Device Manager Software Components

The HiCommand™ Device Manager software product consists of the following basic components:

- **Server.** This document describes and provides instructions for installing and configuring the Device Manager server. The Device Manager server communicates with Hitachi Data Systems 9900 V Series, 9900, 9500 V Series, 9200, and Sun® T3 storage subsystems. In addition, the Device Manager server manages client connections with the Device Manager Web Client and the Device Manager Host Agent(s).

- **Server Security.** Configure the security on your Device Manager server to meet your individual needs. For further information on Device Manager security, see *HiCommand™ Device Manager Server Security Guide* (MK-91HC003).

- **Web Client.** The Device Manager Web Client is a web-based user interface for monitoring and managing Hitachi storage subsystems. The Web Client is a stand-alone Java™-based application that is deployed using the Java Web Start™ software. It communicates with and runs as a client of the Device Manager server. For further information on the Device Manager Web Client, please refer to the *HiCommand™ Device Manager Web Client Installation and Configuration Guide* (MK-91HC001).

- **Command Line Interface (CLI).** The Device Manager CLI enables you to perform client operations by issuing commands from the system command-line prompt. For further information on the command-line interface, please see the *HiCommand™ Command Line Interface Application User Guide* (MK-91HC007).

- **Host Agent.** The Device Manager Host Agent runs on host computers attached to Hitachi Lightning 9900 V Series, Lightning 9900, Thunder 9500V and Thunder 9200 storage subsystems, as well as Sun® Microsystems StorEdge™ T3 subsystems under management by the Device Manager. The Host Agent collects data on the configuration and utilization of the attached storage and sends this information to the Device Manager server. For further information on the Device Manager Host Agent, please refer to the *HiCommand™ Device Manager Agent Installation and Configuration Guide* (MK-92HC019).

1.3 HiCommand™ Suite Common Component

The HiCommand™ Suite common component is a package of features that are used by all HiCommand™ Suite products. The common component is discussed in detail in Chapter 4. Each HiCommand™ Suite product will bundle the common component to use the following functions:

- Single sign-on.
- Integrated logging and repository

Note: Device Manager version 2.3 supports only Tuning Manager version 1.1.

Chapter 2 Windows® Systems Installation

2.1 Overview

This document contains instructions for installing the Device Manager server software on a single central server system. For instructions on installing and using the graphical Web Client software, see the *HiCommand™ Web Client Installation and Configuration Guide* (MK-91HC001).

- Windows® system and media requirements (see section 2.1.1)
- Installing Device Manager (see section 2.2
- Installing InterBase® Server and Client (see section 2.3)
- Verifying Device Manager and common component installation (see section 2.5)
- Upgrading or reinstalling Device Manager (see section 2.6)
- Uninstalling Device Manager (see section 2.7)
- Backing up and restoring the database (see section 2.8)

2.1.1 Windows® System and Media Requirements

The following are the requirements for Windows® systems:

- Operating system: Windows® NT (Service Pack 6 or later) or Windows® 2000 (Service Pack 2 or later). The online help function requires Internet Explorer® 5 or Netscape® 4.76 or higher.

  Note: Windows® 95, Windows® 98, Windows® ME, and Windows® XP are not supported.
- Processor: 300 MHz Pentium II® or better.
- Memory (RAM): 256 MB or more.
- Available hard drive space: 4 GB or more.
- Monitor: VGA with 256 colors or better.
- CD-ROM drive.
- 10/100 Ethernet LAN card.
- Static IP address.
- 9900 V Series, 9900, 9500 V, 9200, or StorEdge™ T3 subsystem and Device Manager server residing on the same network.
- Administrator/root ID to install Device Manager.
- TCP/IP installed and running.

Requirements continue on the following page.

- (9900 and 9900V only) Hitachi SNMP Agent installed and running. This is a program that is installed on the SVP and on each 9900 subsystem. For more information on installing and configuring SNMP, see the following:
  - *9900V Remote Console – Storage Navigator User's Guide* (MK-92RD101).
  - *9900 Remote Console User's Guide* (MK-90RD003).
- Minimum microcode/firmware requirements:
  - 9900 V series microcode level: 21-01-50/00
  - 9900 microcode level: 01-13-19 if not using CVS (Virtual LUN) or LUSE functions, 01-15-39-00/05 if using CVS (Virtual LUN) or LUSE functions
  - 9500 V microcode level: 0651
  - 9200 microcode level: 0559
  - T3 firmware revisions: 1.1.7, 2.0.0
- LAN cables and connections to the subsystem.

Warning: The 9900 and 9900V have a public LAN and a private LAN. Device Manager uses the public LAN to communicate with the SVP about the array and configuration changes. *Do not* under any circumstances attach the private LAN to an external network because this can cause serious problems on the array.

Figure 2.1 illustrates an incorrect LAN connection.

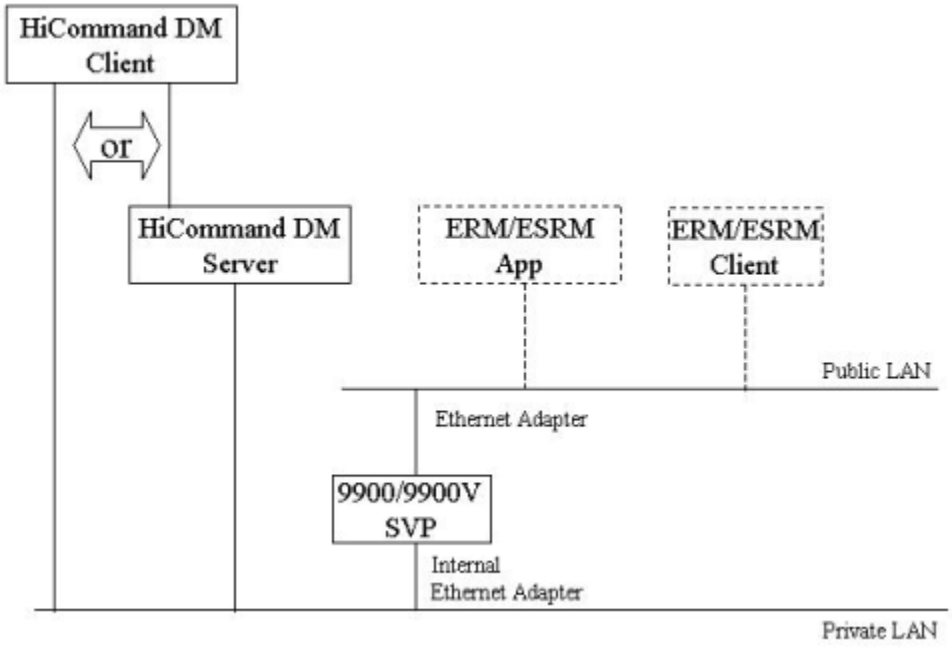

Figure 2.1 Incorrect 9900 and 9900V LAN Connection

For more information on Device Manager requirements for the 9900 V Series, 9900, 9500V, 9200 and T3, please see *HiCommand™ Device Manager Web Client Installation and Configuration Guide* (MK-91HC001).

2.2 Installing the HiCommand™ Device Manager Server and Common Component

Warning: If you are upgrading from Device Manager version 2.2 or earlier to Device Manager 2.3 or later, the directory structure has changed. You need to delete the earlier version (see section 2.7 for instructions), and then do a clean install of the later version.

Warning: The default installation directory has changed. For versions 2.2 and earlier, Device Manager was installed in **c:\program files\HiCommand**. For versions 2.3 and later, Device Manager will be installed in **c:\program files\HiCommand\Device Manager**, and the common component will be installed in **c:\program files\Hitachi\hntrlib2.**

Note: Always read the release notes before installing Device Manager, and make sure that you have the minimum required microcode version on the subsystem.

The Device Manager server installation CD includes the following applications:

- Java2™ JRE: Version 1.3.1_03 (transparent to the user)
- HiCommand™ Suite common component software (automatically installed during the Device Manager installation.)
- Device Manager server software (described in this section)
- InterBase® and InterClient® systems: Version 6.0.1 Server and Client software (see section 2.3)

Once these products are installed, the Device Manager client software can be downloaded, installed, and updated on the client system from the Device Manager server.

Before you begin, verify the following:

- The platform running Device Manager has no other applications using the standard SNMP ports (161 and 162).

  Note: Because CA Unicenter TNG® and HP CommandView® systems sometimes use these ports, verify that these systems are not active.

- No other applications are running.
- The user account is a member of the Windows® Administrator group.

Note: If you do not have the current versions of InterBase® Server/Client and the InterClient® Driver for InterBase® software installed, please review section 2.3 before installing Device Manager.

To install the Device Manager Server:

1. Log on to the system as an administrator.
2. Stop any other HiCommand™ Suite software that is running.
3. Stop the single sign-on service and the common web service from the Windows® Services Panel (see Figure 2.2 for the Windows® 2000 Services panel, and Figure 2.3 for the Windows NT® Services panel). On the Services panel, select **HiCommand Server → Stop.**

   Note: To access the Services panel in Windows® 2000, select **Start → Settings → Control Panel → Administrative Tools\Services**. In Windows NT®, select **Start → Settings → Control Panel → Services**.
4. Load the Device Manager installation CD-ROM. The CD should launch automatically. If it does not launch, from the **Start** menu, select **Run → Browse**. Navigate to the CD directory and select **install.exe** from the list of root files.
5. The Introduction panel (see Figure 2.4) displays. Select **Next** to continue.
   Note: If you see the HiCommand Suite is running panel (see Figure 2.5), you must stop all of the HiCommand™ Suite software before installation.
6. If lower (unsupported) versions of InterBase® and InterClient® software are already installed, the Incorrect InterBase Version panel (see Figure 2.6) and Incorrect InterClient Version panel (see Figure 2.7) display. The installer halts at this point to allow you to uninstall these previous software versions. Select any key to exit the installation. See section 2.7.2 for instructions on how to uninstall InterBase® Server and Client, and section 2.7.3 to uninstall InterClient®. Once you have uninstalled the unsupported versions, re-run the installer.
7. If InterBase® Server is not installed, the InterBase Not Installed panel (see Figure 2.8) displays. Select **Next** to continue. If InterClient® is not installed the InterClient Not Installed panel (see Figure 2.9) displays. Select **Next** to continue. InterBase® Server and InterClient® are installed as part of the overall installation (see sections 2.3.2 and 2.3.3). ***Note***: If these panels do not display, InterBase® Server and InterClient® will not be installed.
8. If the common component is not installed, HiCommand Suite Common Component is Not Installed panel displays (see Figure 2.10). Select **Next** to continue.

Instructions continue on the following page.

9. The License Agreement panel (see Figure 2.11) displays. Select **I Accept** and then **Next**. The Choose Install Folder panel (see Figure 2.12) displays.

10. To accept the default folder, select **Next**. If you want to change to another install folder, make that selection and then select **Next**.

11. The Install Folder Information panel displays. You will see either Figure 2.13 (if you have not previously installed the HiCommand™ common component, or Figure 2.14 (if the common component is installed).

12. ***Note***: The default installation directory has changed. For versions 2.2 and earlier, Device Manager was installed in **c:\program files\HiCommand**. For versions 2.3 and later, Device Manager will be installed in **c:\program files\HiCommand\Device Manager**, and the common component will be installed in **c:\program files\Hitachi\hntrlib2.** Select **Next** to continue**.**

13. The Pre-Installation Summary panel (see Figure 2.15) displays. Verify that the information is correct and select **Install**.
    Warning: Do not cancel after you select **Install**.

14. ***Note***: If you see the message "HiCommand™ Suite common component Error", look in the HiCommand™ Device Manger Install log file and follow the instructions. In addition, you must manually remove the installation directory, reboot the system, and reinstall Device Manager.

15. The **InterBase Server Setup** panel may appear (see Figure 2.16). See section 2.3 for the installation instructions for InterBase® and InterClient®.

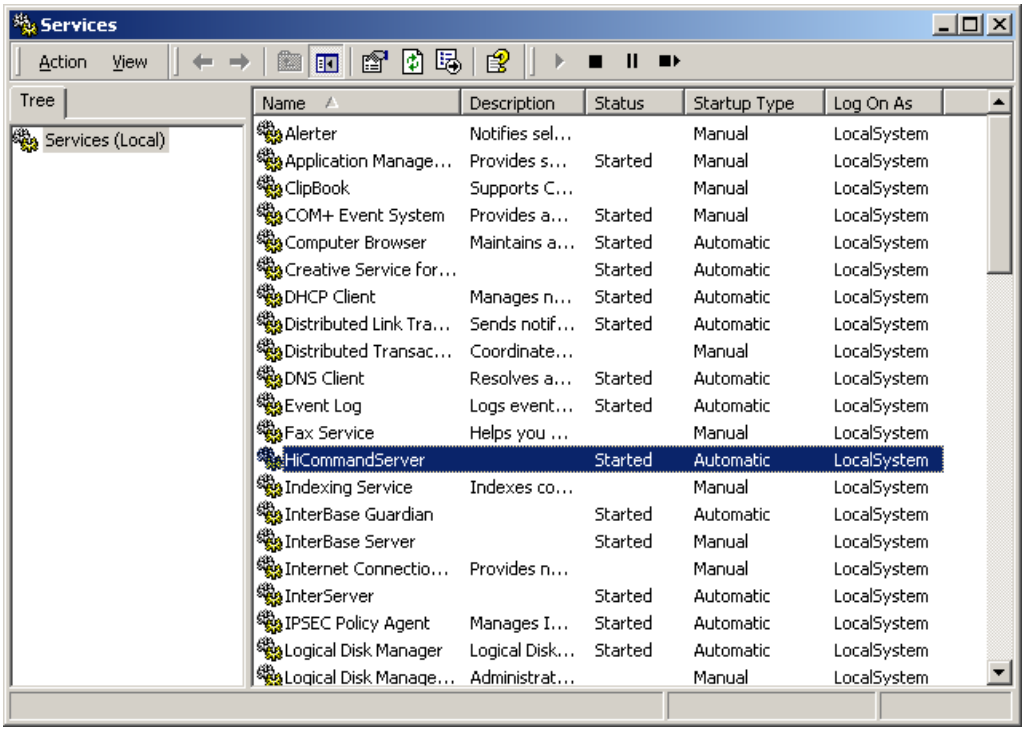

Figure 2.2 Services Panel - Windows® 2000

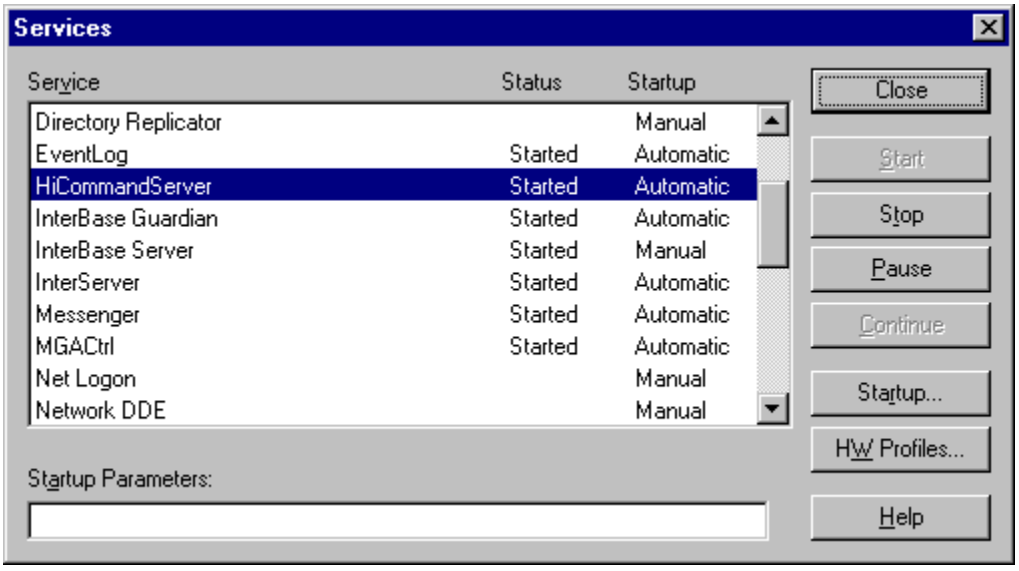

Figure 2.3 Services Panel - Windows NT®

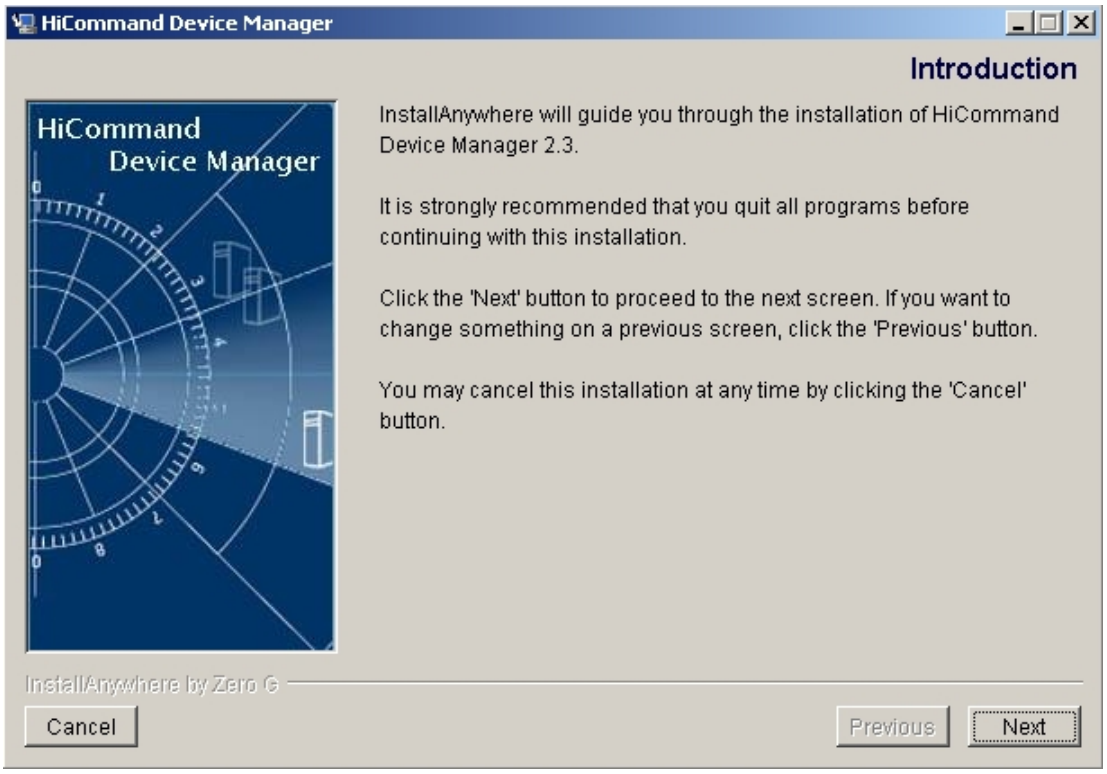

Figure 2.4 Introduction Panel

Figure 2.5 HiCommand Suite Is Running Panel

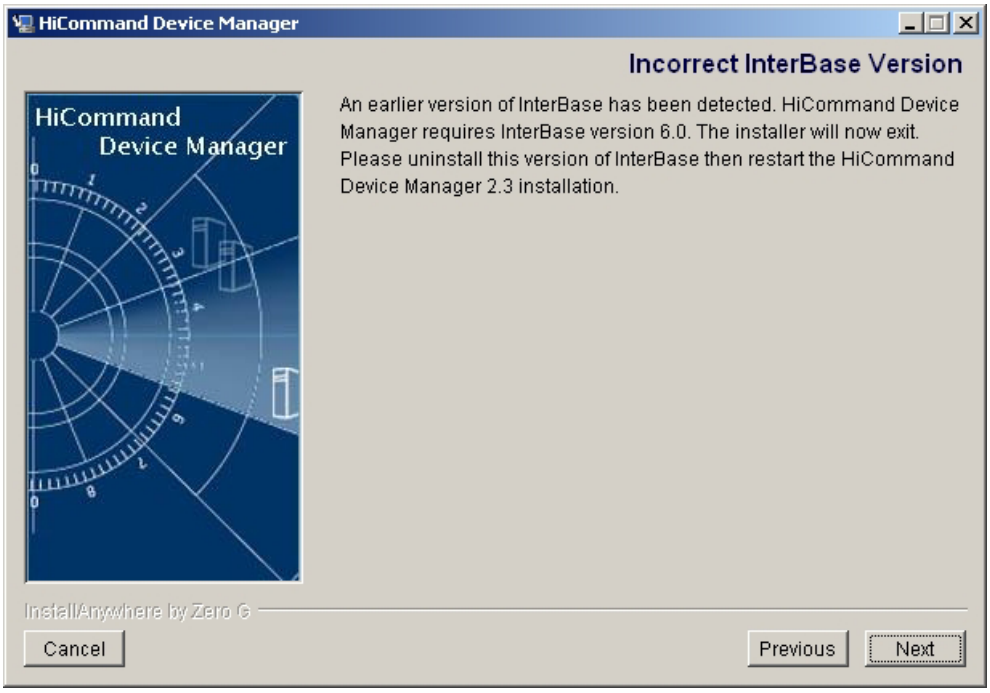

Figure 2.6 Incorrect InterBase Version Panel

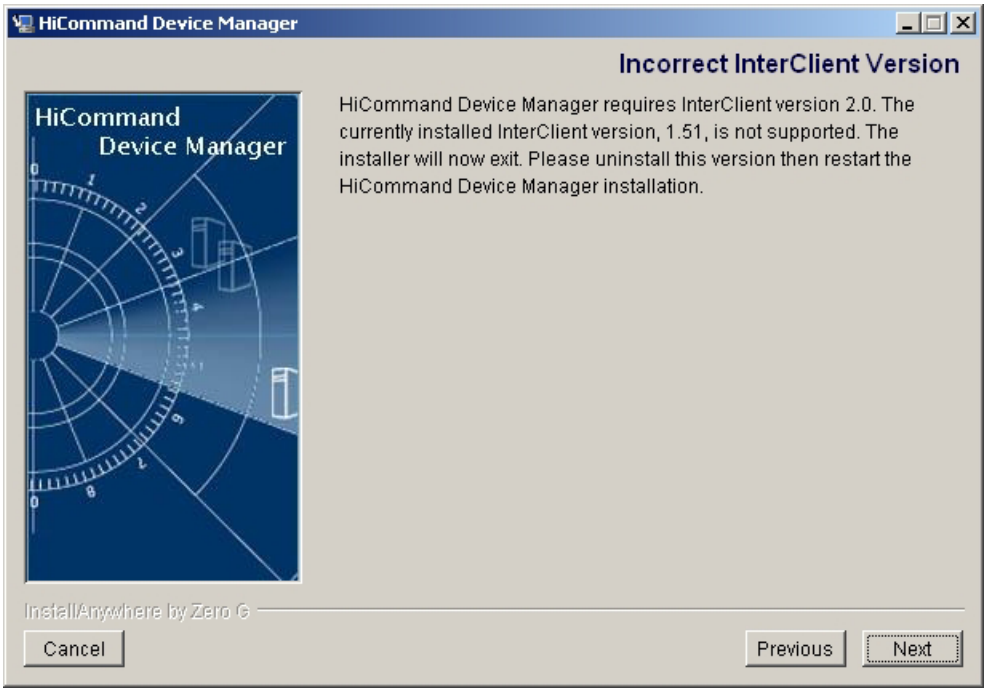

Figure 2.7 Incorrect InterClient Version Panel

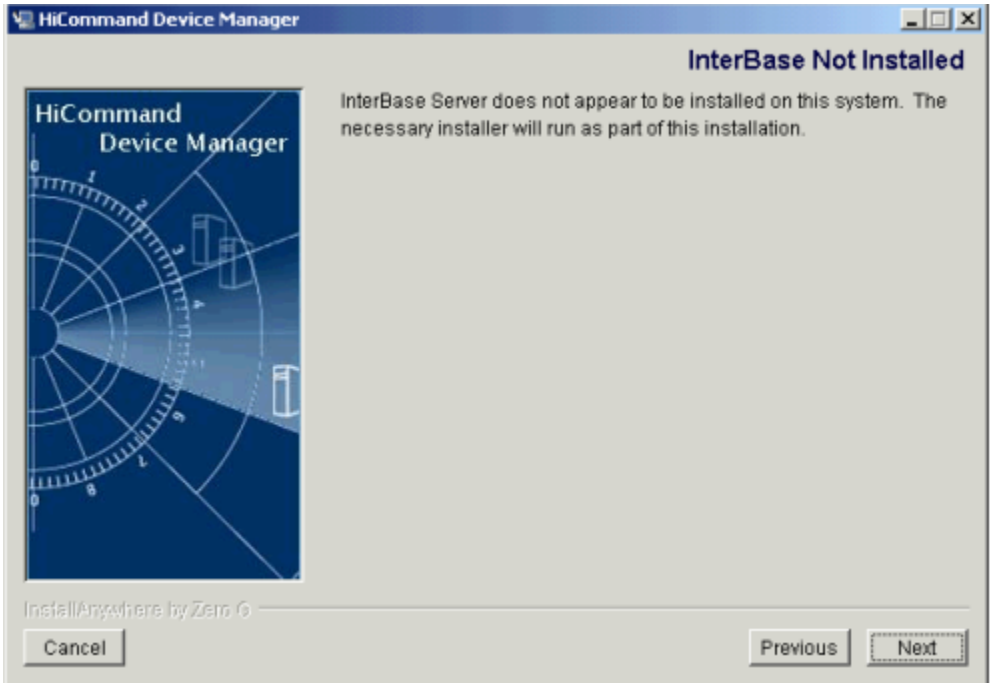

Figure 2.8 InterBase Not Installed Panel

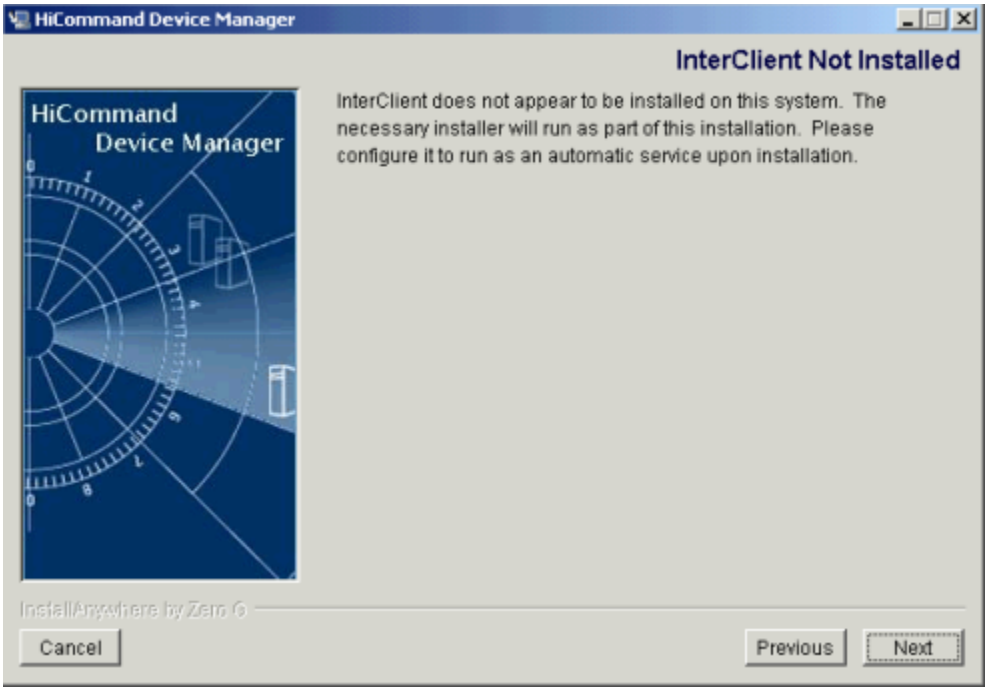

Figure 2.9 InterClient Not Installed Panel

Figure 2.10 HiCommand Suite Common Component Not Installed Panel

Figure 2.11 HiCommand Server License Agreement Panel

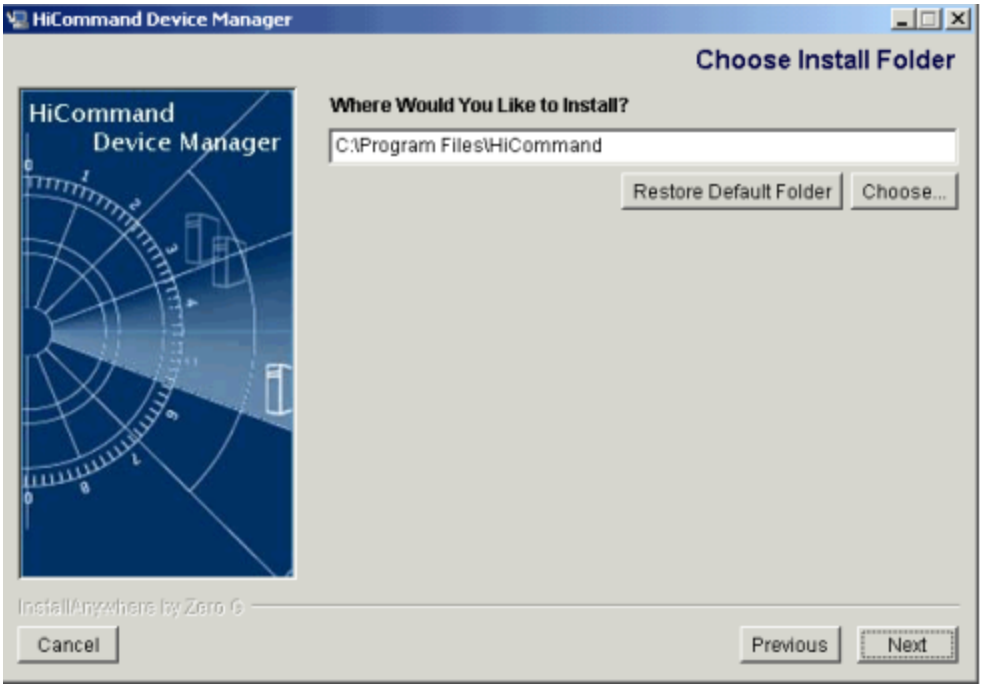

Figure 2.12 Choose Install Folder Panel

Figure 2.13 Install Folder Information Panel (HiCommand™ Suite Common Component Not Installed)

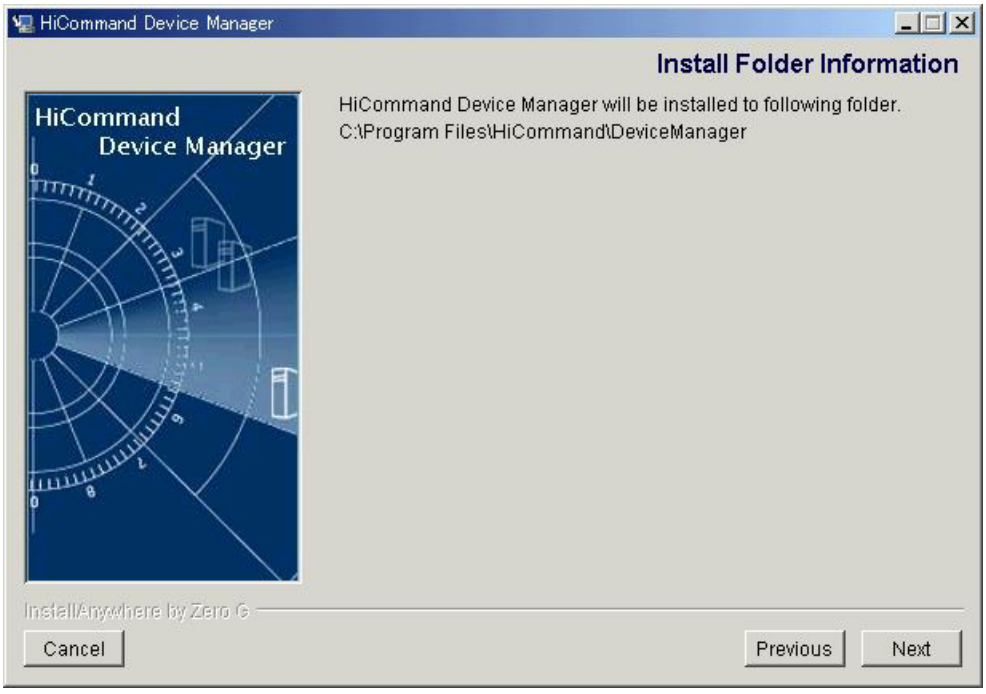

Figure 2.14 Install Folder Information Panel (HiCommand™ Common Component Already Installed)

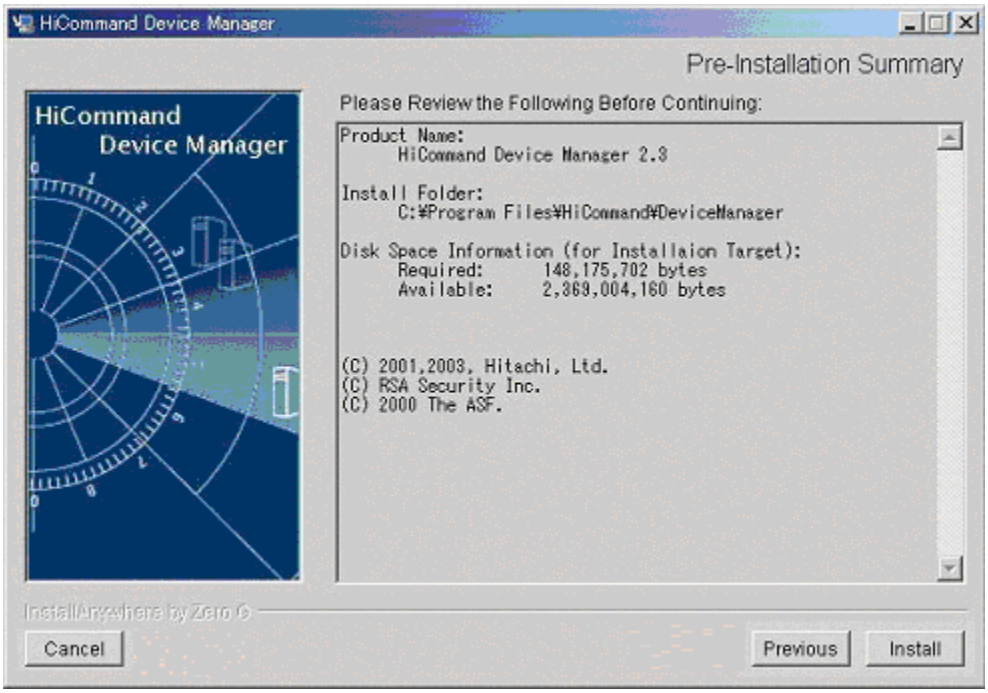

Figure 2.15 Pre-Installation Summary Panel

2.3 Installing InterBase® Server, InterBase® Client, and InterClient®

The Device Manager requires InterBase® Server and Client version 6.0 and the InterClient® 2.0 JDBC Driver for InterBase®. If you have already installed these products at the specified version levels or higher, this part of the installation process is automatically skipped.

2.3.1 Overview of InterBase® Installation

Note: There are actually three InterBase® services that are installed on Windows® systems. In addition to the InterBase® Server and InterServer® services referenced in this section, InterBase® Guardian is also installed. The InterBase® Guardian service monitors the InterBase® Server service, and restarts the service if necessary. The InterBase® Guardian service is not required to run the Device Manager, but it does offer extra protection.

The InterBase® database server used by Device Manager periodically creates and uses temporary files. The InterBase® software creates these temporary files in a directory specified by the InterBase® configuration file or by environment variables.

- A specified temporary directory must exist.
- The hard drive must have 20 MB of free space.
- If no entry is found in the configuration file, and no **INTERBASE_TMP** environment variable exists, InterBase® uses the directory pointed to by the system's **TMP** environment variable.

  You can use the default directories, or specify other directories. The default directories are:

  - Windows® 2000: **C:\WINNT\TEMP**
  - Windows NT® 4: **C:\TEMP**
- You can also specify that the InterBase® software use directories other than the default hard-coded directories, using one of the following methods:
  - The InterBase® configuration file is in the InterBase® directory. On Windows® systems, the filename is **ibconfig**. You can create an entry in this file in the following format:

    TMP_DIRECTORY size "pathname"
  - You can also create the environment variable **INTERBASE_TMP** which points to the directory you would like the InterBase® software to use for temporary files.

2.3.2 Installing InterBase® Server and Client

To install the InterBase® Server and Client:

1. If you have not already installed InterBase® Server and Client, the **InterBase Server Setup** panel (see Figure 2.16) displays within the Windows® installation process.
2. Select **Next** to open the **InterBase Server Information** panel (see Figure 2.17). Reading the entire contents of this panel is strongly recommended.
3. Select **Next** to open the **InterBase® Software License Agreement** (see Figure 2.18). Select **Yes** to continue with the installation (or **No** to cancel).
4. When the **InterBase Component Selection** panel (see Figure 2.19) displays, make sure that all components are checked. Select **Install**. When the InterBase® installation is complete, the **InterBase Sever Setup Complete** panel (see Figure 2.21) displays.
5. ***Note***: The Information panel (see Figure 2.20) displays if certain files are in use, warning that InterBase® Guardian and InterBase® Server will not run until the server is restarted.
6. Select **Finish** to complete the installation.

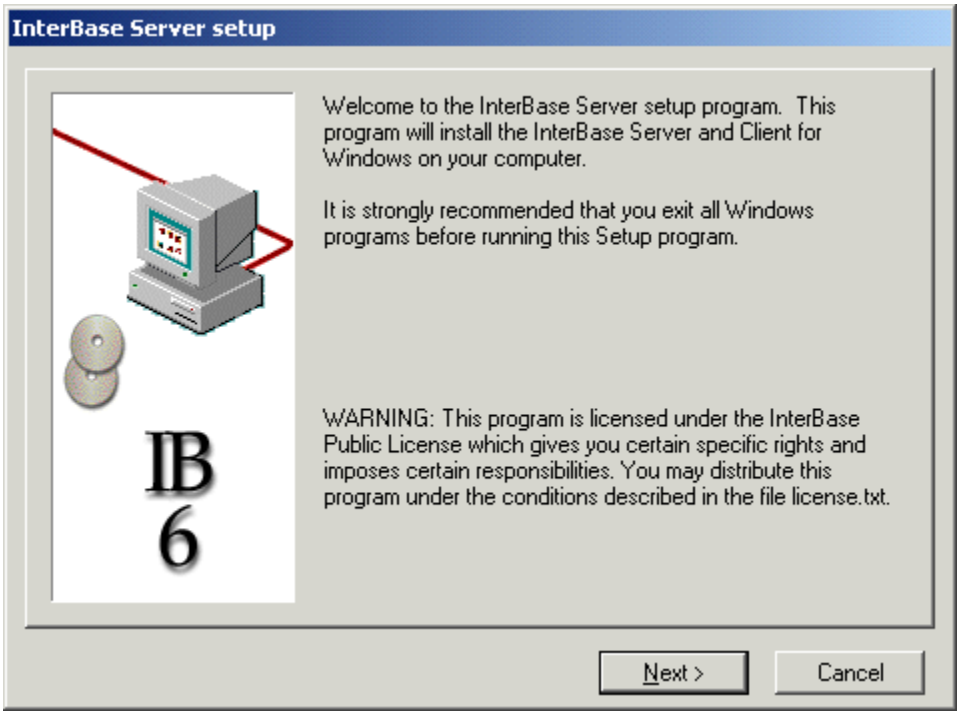

Figure 2.16 InterBase Server Setup Panel

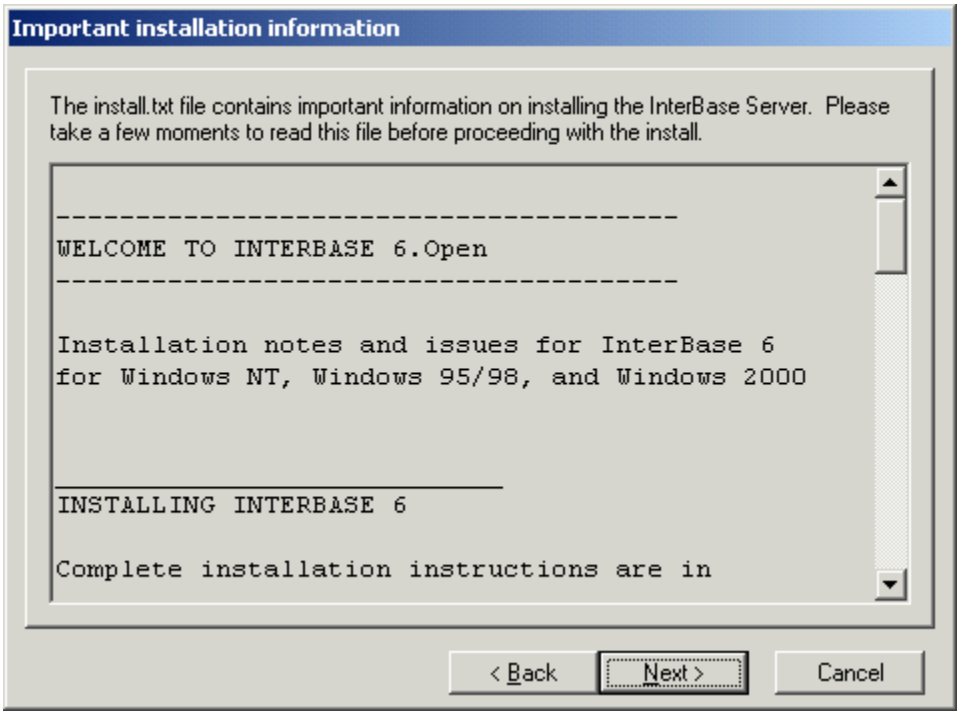

Figure 2.17 InterBase Installation Information Panel

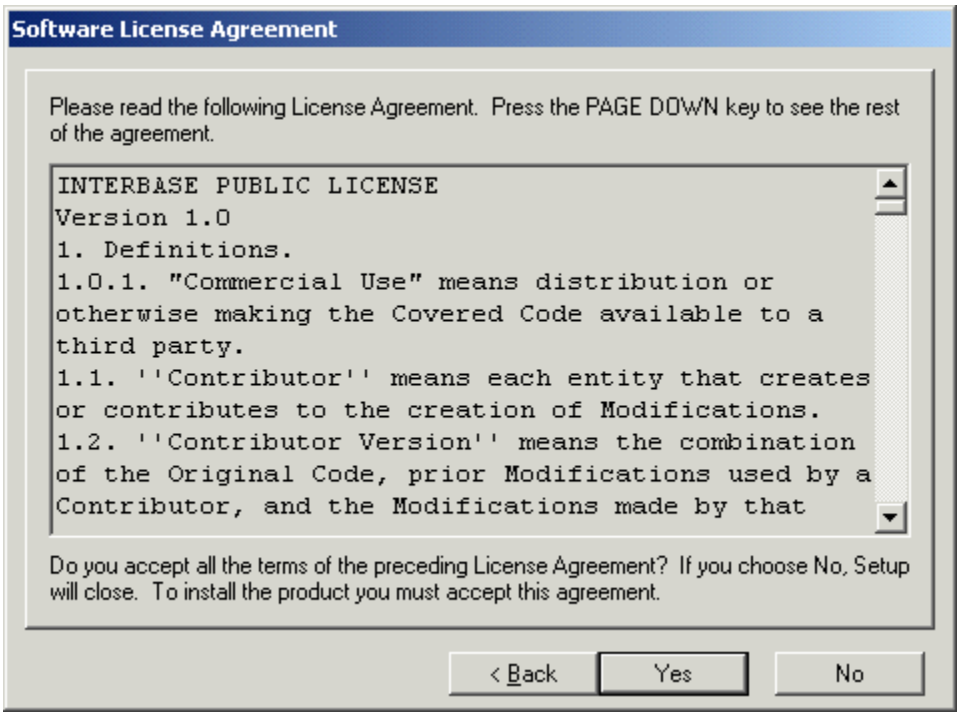

Figure 2.18 InterBase Software License Agreement Panel

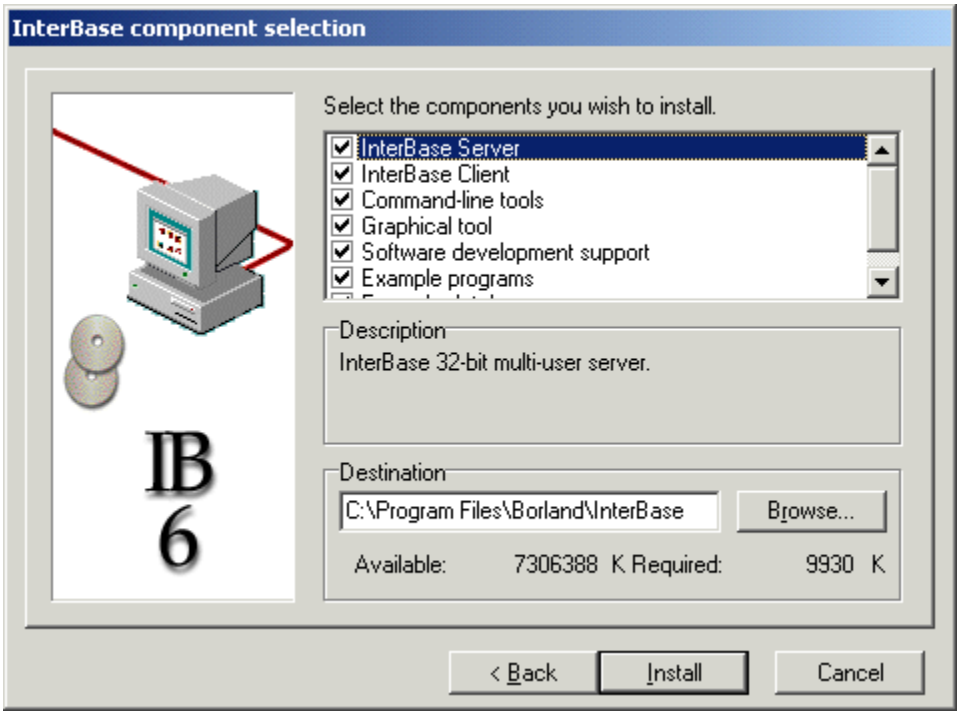

Figure 2.19 InterBase Component Selection Panel

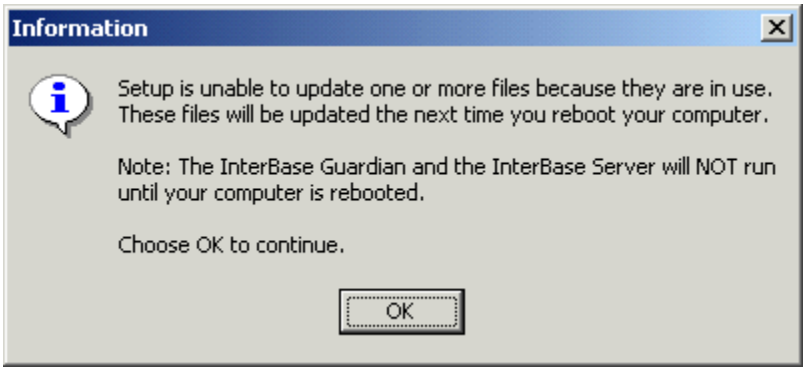

Figure 2.20 InterBase Warning Panel

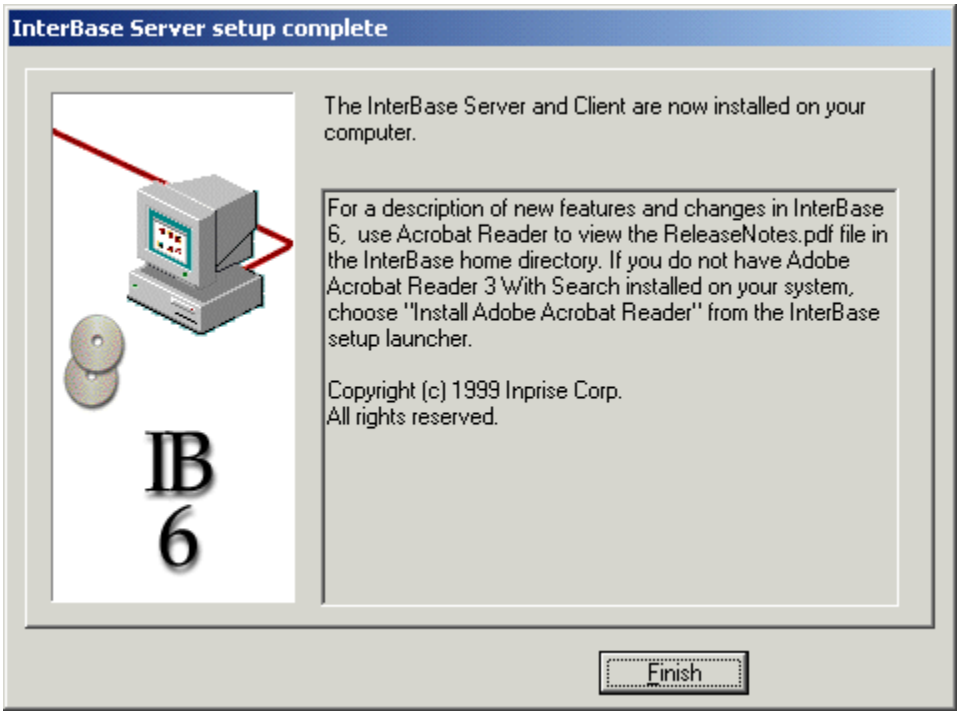

Figure 2.21 InterBase Server Setup Complete Panel

2.3.3 Installing InterClient®

To install InterClient® during the Windows® installation process:

1. Select **Yes** on the InterClient Self-Extracting EXE panel (see Figure 2.22) to continue the installation. The InterClient Installation Setup panel (see Figure 2.23) displays.
2. Select **Next** to continue. The InterBase Public License panel (see Figure 2.24) displays.
3. Select **Yes** to continue (or **No** to cancel). The InterClient Installation Notes panel (see Figure 2.25) displays. ***Important***: These notes contain updated information.
4. Select **Next** to continue. The InterClient Special Notes panel (see Figure 2.26) displays.
5. Select **Next** to continue. The InterClient Select Components panel (see Figure 2.27) displays. Make sure that all components are selected.
6. Select **Next**. The InterServer Installation panel (see Figure 2.28) displays.
7. The InterBase® software should already be installed, so select **Next** to continue.
8. If the **Modifying TCP/IP Services** panel displays, select **Let Setup modify the SERVICES file**. Select **Next**. The InterClient InterServer Configuration panel (see Figure 2.31) displays.
9. ***Note***: If this is the first time that the InterServer® software has been installed on this system, the Modifying TCP/IP Services panel (see Figure 2.29) is displayed. This panel does not display if this is an upgrade or a reinstallation.
10. Select **Configure InterServer**, then select **Next** to continue.
11. Before the files are copied, the Start Copying Files panel (see Figure 2.30) displays. Review the current settings. Select **Install** to continue. The InterServer Configuration Utility panel (see Figure 2.32) displays.
12. In the **Server Startup** field, from the drop-down list, select **Service**. In the **Startup Mode** field, from the drop-down list, select **Windows Startup**. Select **OK** to continue.

    Warning: The Interserver® software *must* be configured to start automatically as a service or it does not start properly and HiCommand™ is not launched.
13. If the Unknown Error panel (see Figure 2.33) displays, there is an unrecoverable error in the InterServer® installation. Uninstall and reinstall the InterServer® software.
14. If the InterServer Error panel (see Figure 2.34) displays, you have not logged in as an administrator-level user. Shut down and re-log in as an administrator. (If this panel displays and you have already logged in as an administrator-level user, ignore it.)
15. Select **Yes** on the View Readme File panel (see Figure 2.35)to view the recommended readme file (or **No** to continue). The Setup Complete panel (see Figure 2.36) displays.
16. The Information panel (see Figure 2.37) displays, warning you that the Explorer Shell Start Menu might not be updated immediately. This is normal. Select **OK** to continue.

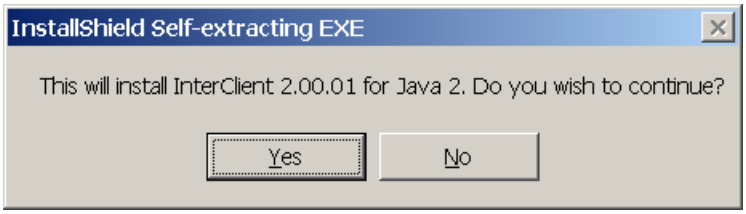

Figure 2.22 InterClient Self-Extracting EXE Panel

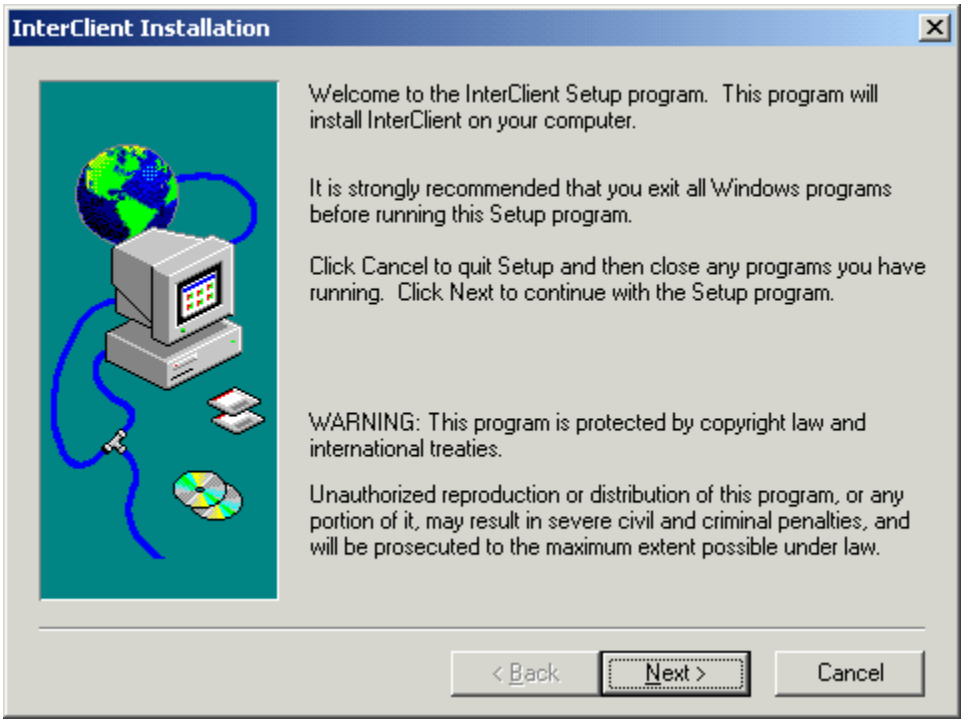

Figure 2.23 InterClient Installation Setup Panel

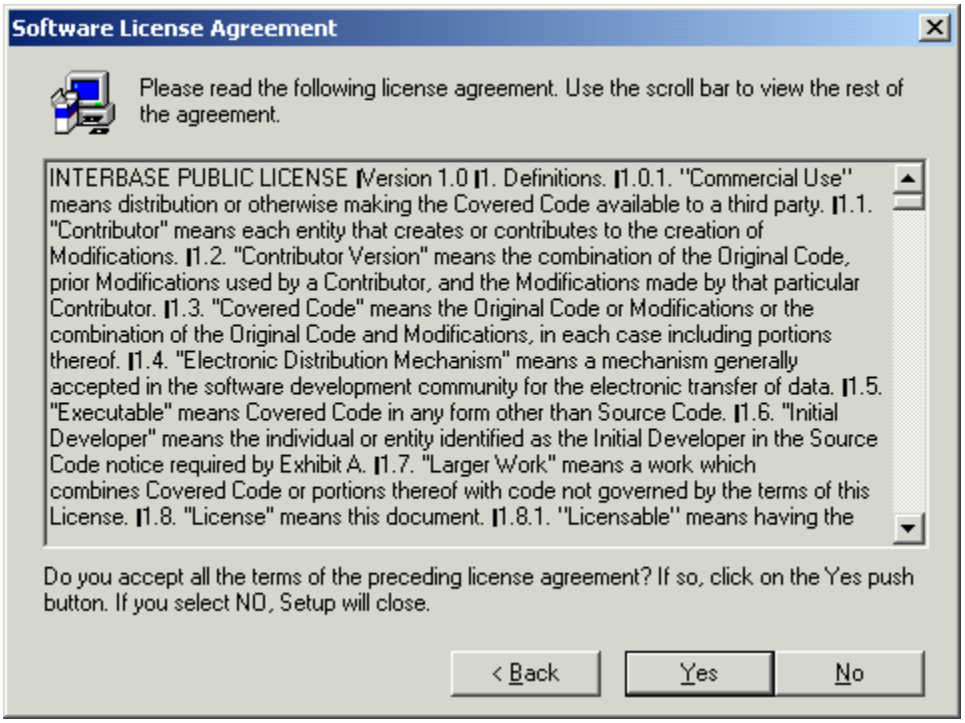

Figure 2.24 InterBase Public License Panel

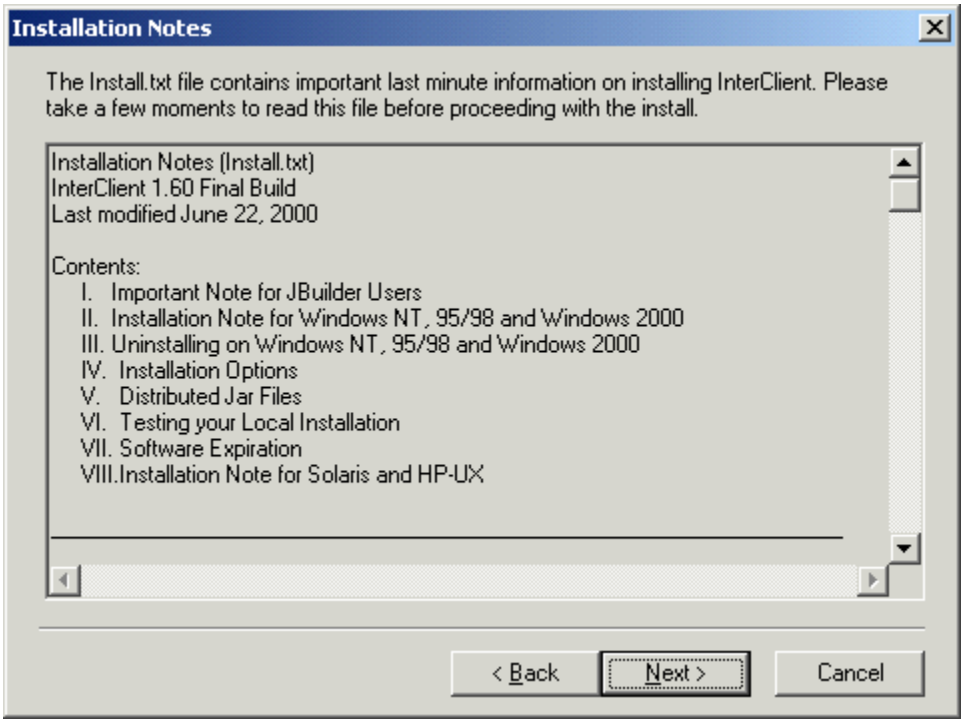

Figure 2.25 InterClient Installation Notes

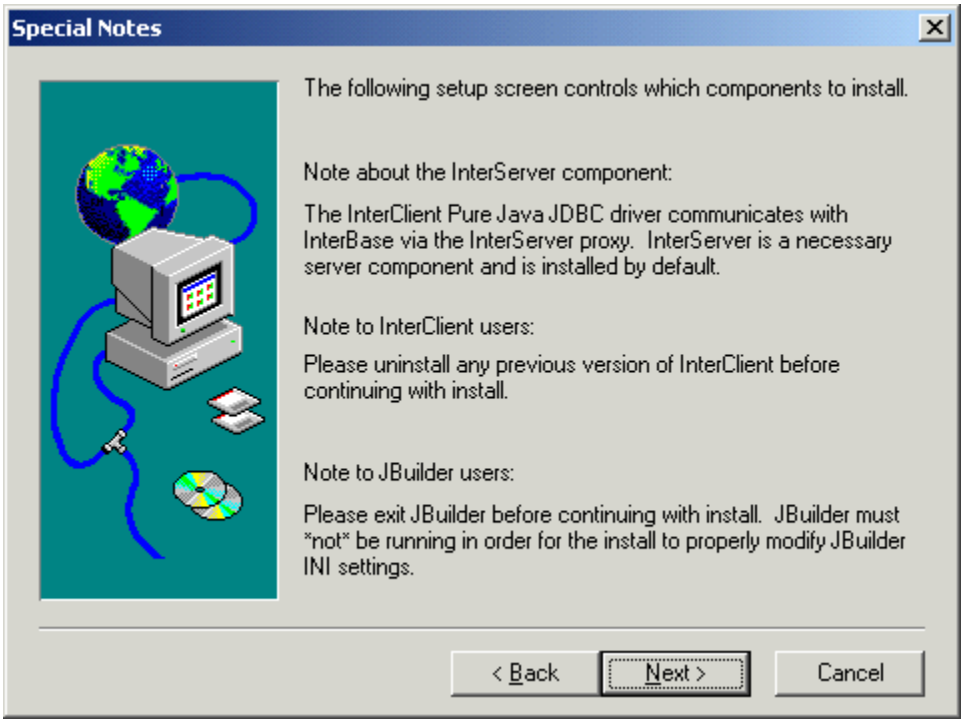

Figure 2.26 InterClient Special Notes Panel

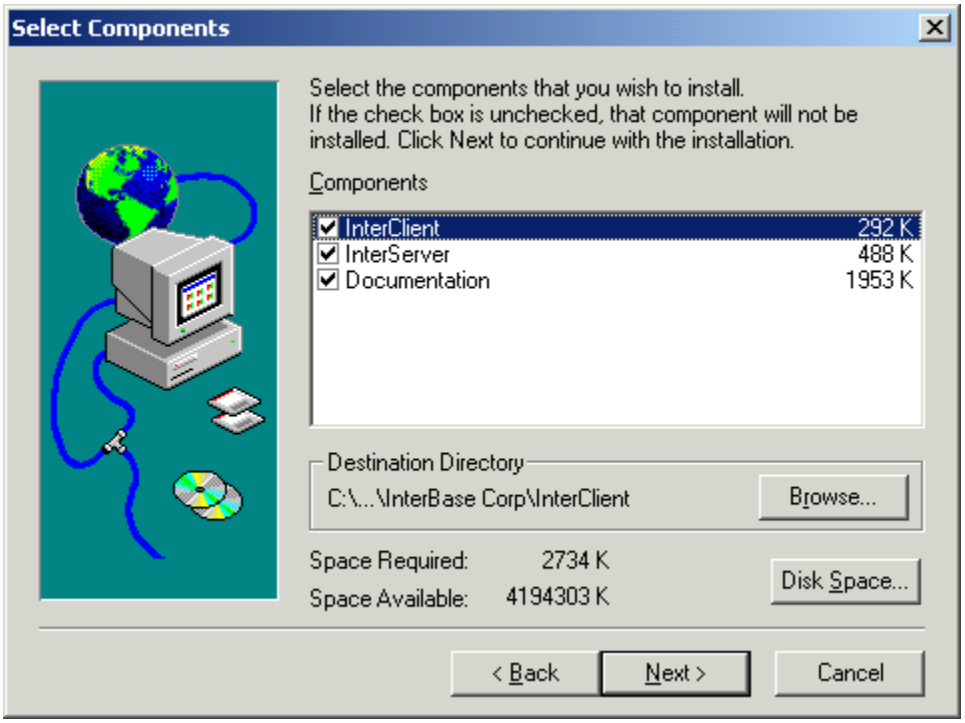

Figure 2.27 InterClient Select Components Panel

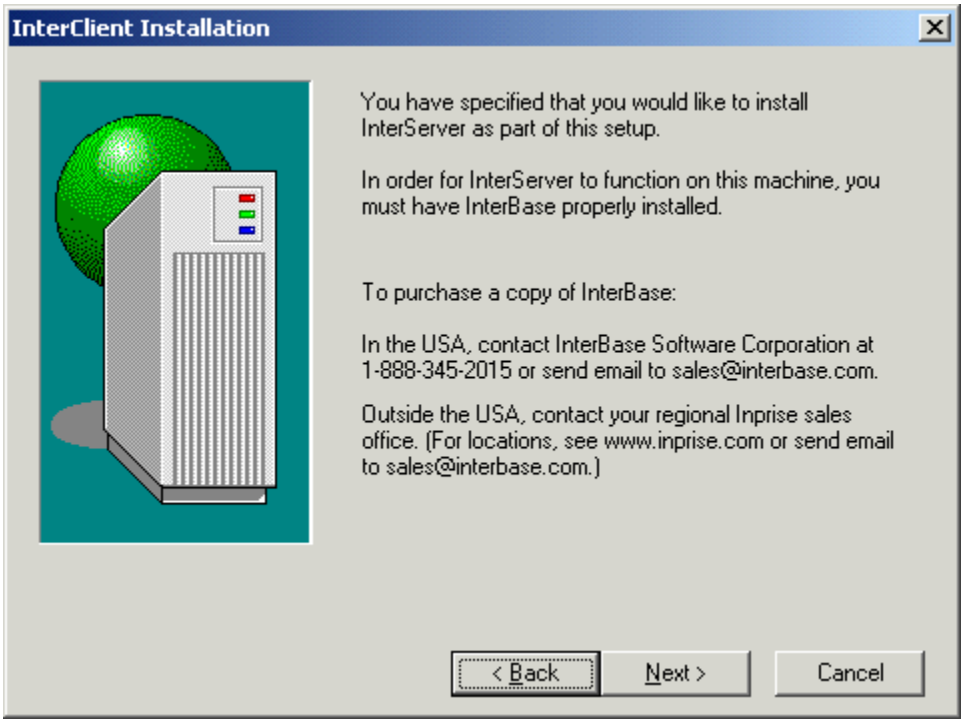

Figure 2.28 InterServer Installation Panel

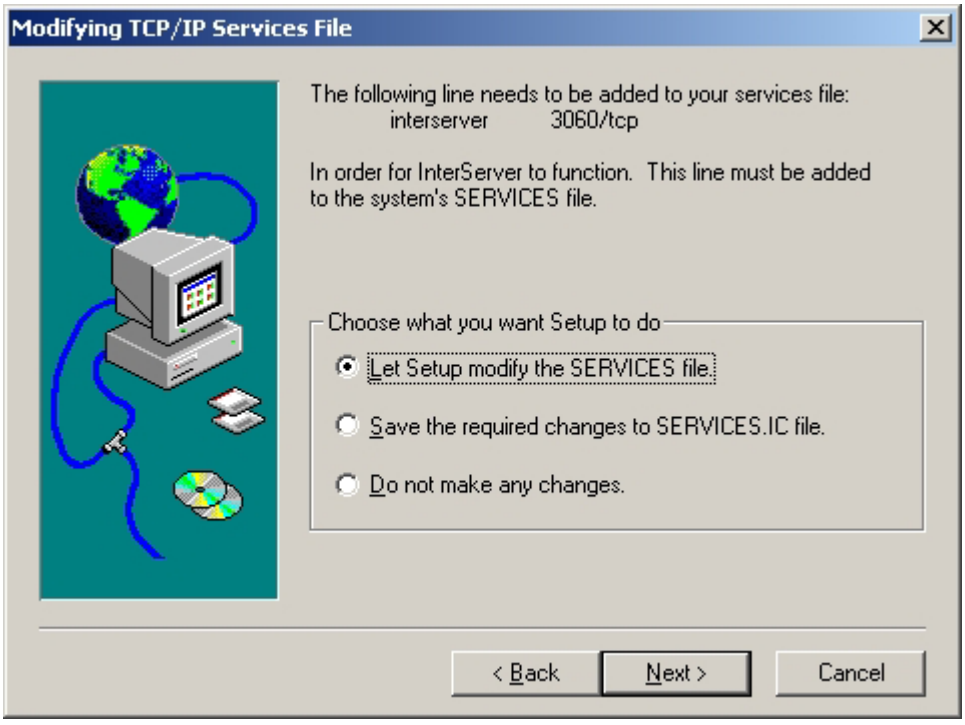

Figure 2.29 Modifying TCP / IP Services File Panel

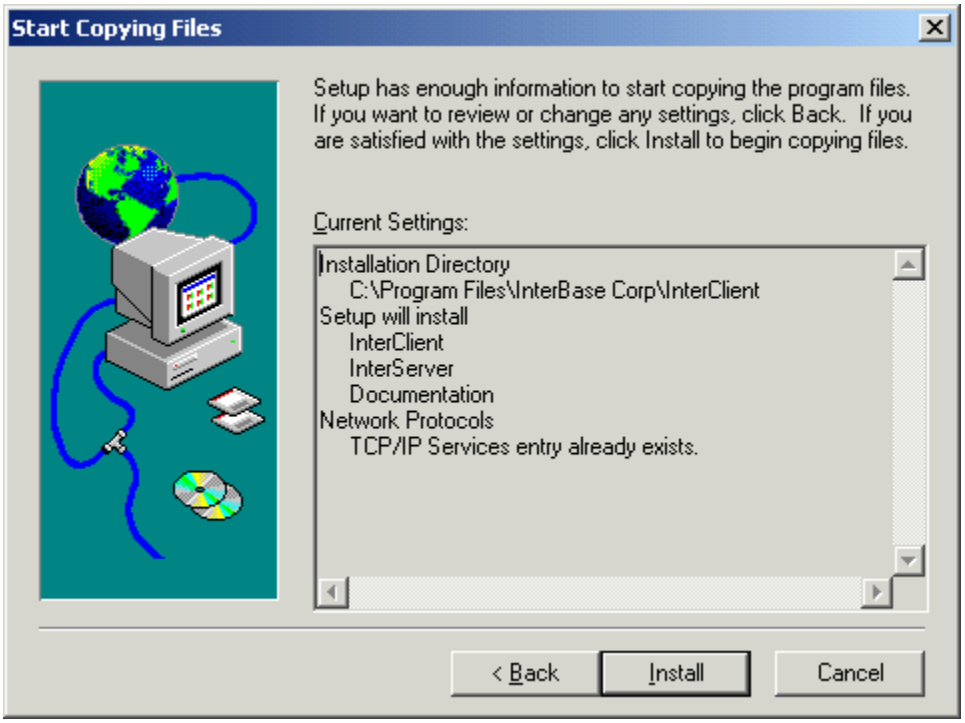

Figure 2.30 Setup Overview

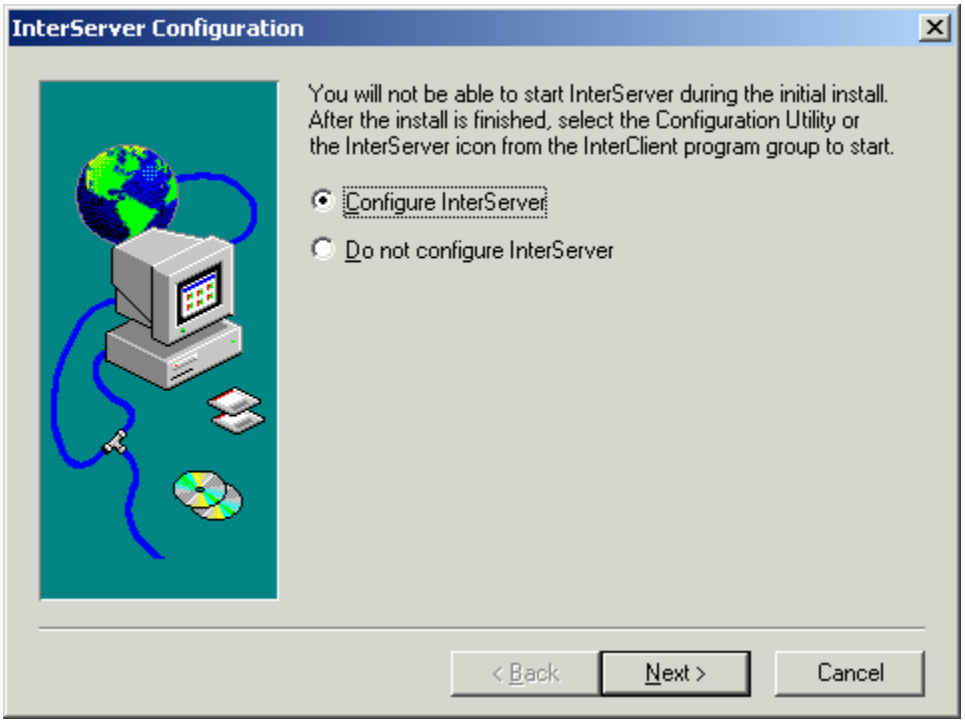

Figure 2.31 InterServer Configuration Panel

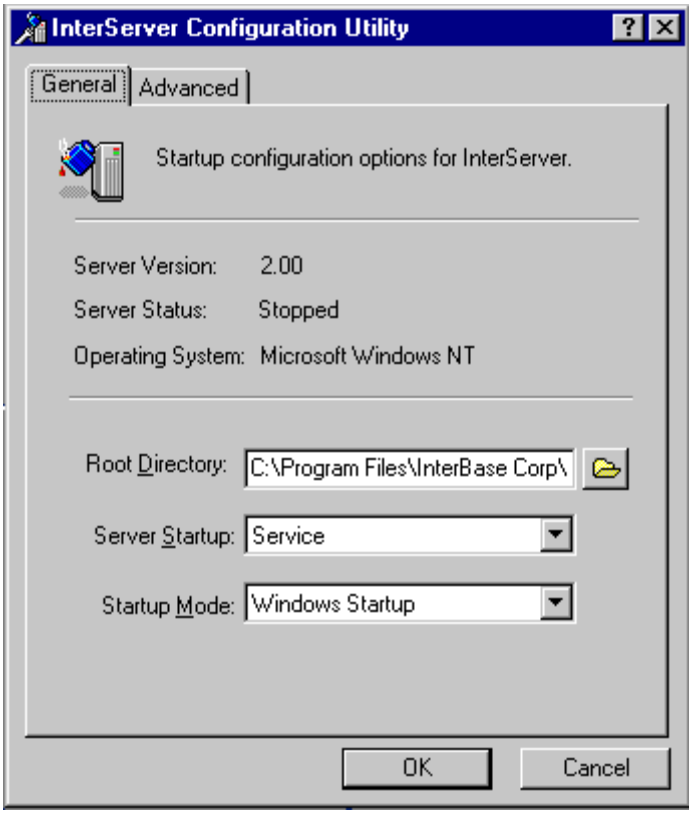

Figure 2.32 InterServer Configuration Utility Panel

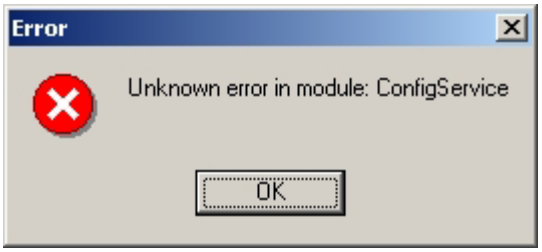

Figure 2.33 InterServer Unknown Error Panel

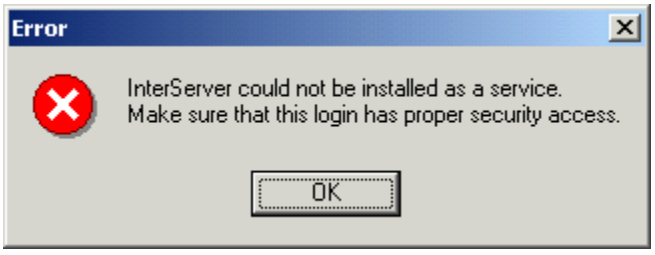

Figure 2.34 InterServer Error Panel

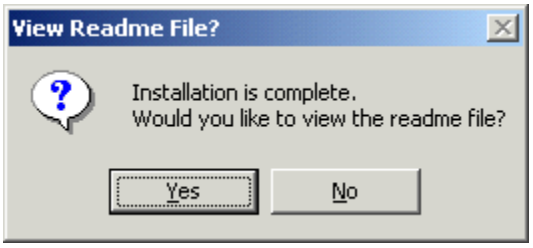

Figure 2.35 View Readme File Panel

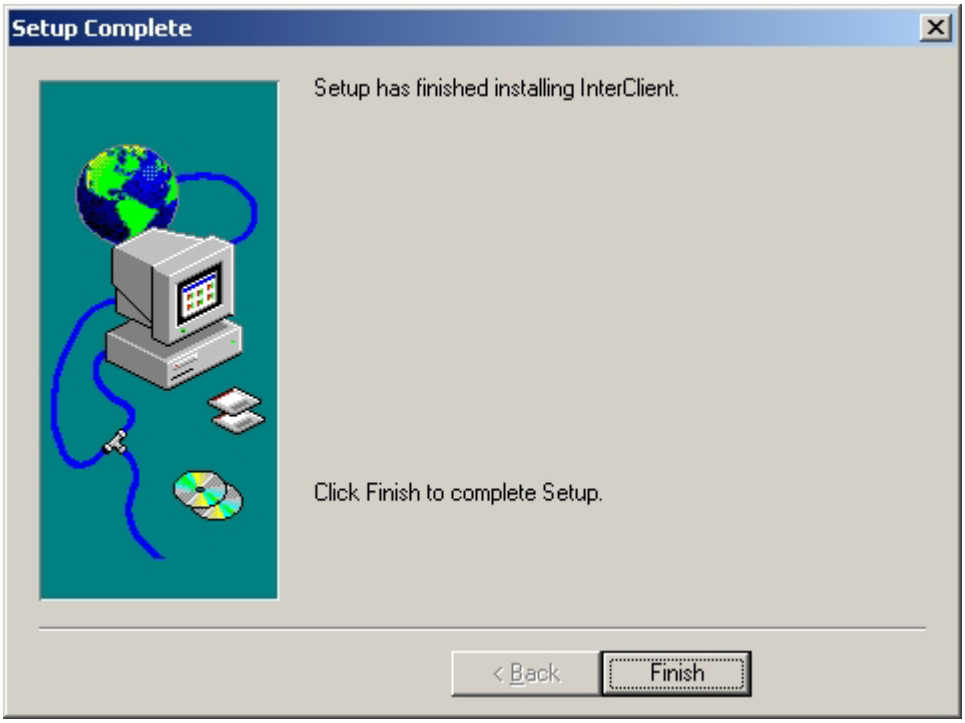

Figure 2.36 Setup Complete Panel

Figure 2.37 Information Panel

2.4 Completing the Installation

Whether or not InterBase® or InterClient® were installed with the Device Manager server, the installation will complete in the following manner:

1. The Secure Socket Certificates Note panel (see Figure 2.38) displays. Select **Next** to continue.
2. The Start Services panel (see Figure 2.39) displays, reminding you to verify that the InterBase® Server, InterServer®, the single sign-on service, the common web service, and the Device Manager server are all started. A reboot is not required if you start the services manually.
3. To verify that these services have been started, open the Services panel (refer to Figure 2.2 for the Windows® 2000 Services panel, and Figure 2.3 for the Windows NT® Services panel).
   Note: To access the Services panel in Windows® 2000, select **Start → Settings → Control Panel → Administrative Tools\Services**. In Windows NT®, select **Start → Settings → Control Panel → Services**.
4. If you need to start these services manually, you may do so from this panel. Once you have verified that the required services have been started, select **Next** on the Start Services panel (see Figure 2.39) to continue.
5. The Install Complete panel (see Figure 2.40) displays.
6. Select **Finish** to complete the installation.

Figure 2.38 Secure Socket Certificates Note Panel

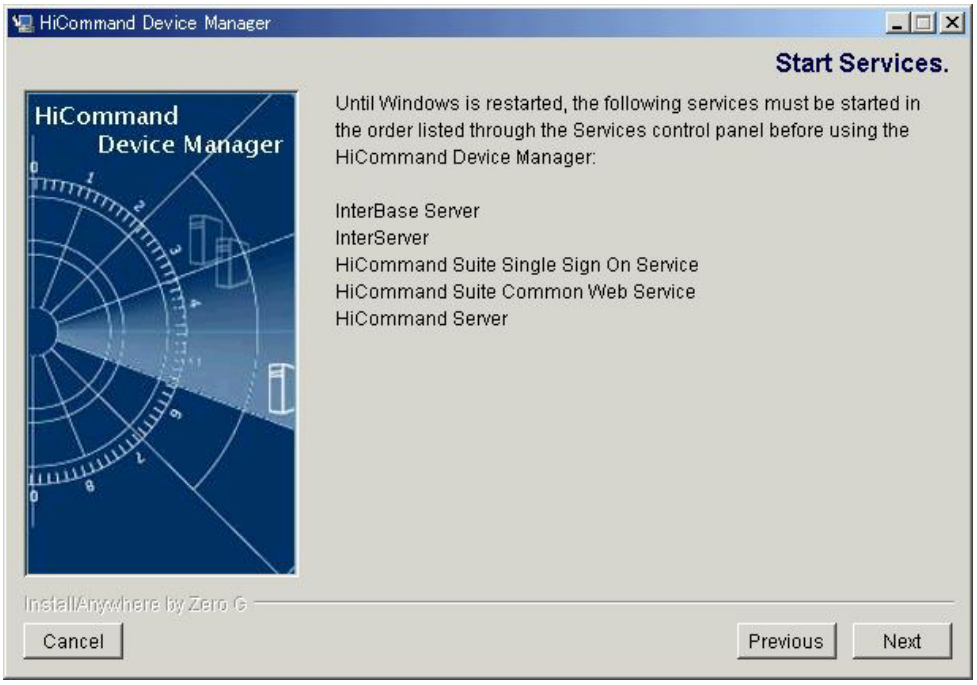

Figure 2.39 Start Services Panel

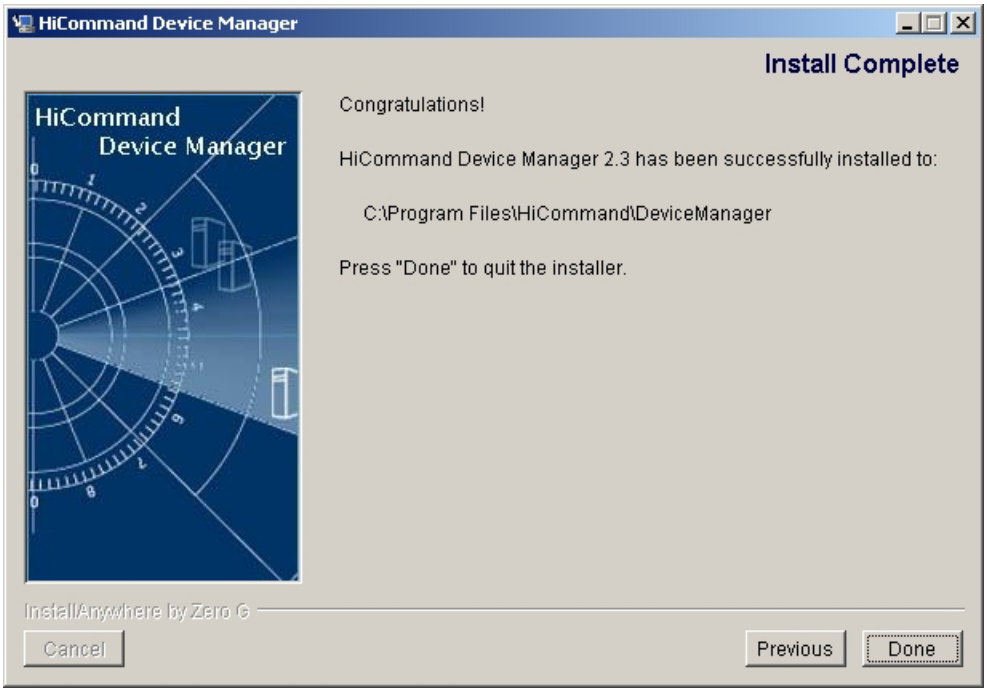

Figure 2.40 Install Complete Panel

2.5 Verifying Device Manager Server and Common Component Installation

After you complete the installation, you need to verify that the following services are either running or can be started manually:

- InterBase® server
- InterServer®
- HiCommand™ Suite single sign-on service
- HiCommand™ Suite common web service
- HiCommand™ server

If all of these services can be started, the installation was successful. If one or more of these services either does not appear in the Services panel (refer to Figure 2.2 for Windows® 2000 or Figure 2.3 for Windows NT®) or cannot be started, the installation has failed.

2.5.1 InterServer® Does Not Appear in the Services Panel

If InterServer® does not appear in the Services panel (refer to Figure 2.2 for Windows® 2000 or Figure 2.3 for Windows NT®), the InterClient® installation and configuration may have failed.

1. Select **Start → InterBase InterClient → InterServer Configuration Utility** to open the InterServer Configuration panel (refer to Figure 2.31). Select the **General** tab, then choose **Service** for Server Startup, and **Windows® Startup** for the Startup Mode (refer to Figure 2.32).
2. Click **OK**, and start the InterServer® service from the Services Panel.

2.5.2 InterServer® Does Not Start

If InterServer® does not start, the installation and configuration may have failed.

1. Uninstall InterClient®. Refer to section 2.7.3 for instructions.
2. Reboot the system.
3. To reinstall InterClient®, select **<installation directory>\InterBaseInstallers\InterClient\ IC20001winJRE12.exe**. Refer to section 2.3.3 for installation instructions.
4. Start the InterServer® service from the Services panel.

2.5.3 InterBase® Server Does Not Start, or Does Not Appear in the Services Panel

If the InterBase Server® either does not appear in the Services panel (refer to Figure 2.2 for Windows® 2000 or Figure 2.3 for Windows NT®) or does not start, the installation and configuration may have failed.

1. Verify InterBase® installation by checking the Windows Add/Remove programs panel.
2. If InterBase® is installed, you need to uninstall it. See section 2.7.2 for instructions.
3. Reboot the system.
4. Unzip **<Installation directory>\InterBaseInstallers\InterBase_WI-V6.0.1-server.ZIP**, and install InterBase®. Refer to section 2.3 for installation instructions.
5. Copy **<Installation directory>\HiCommandServer\database\interbase\HiCommand.gbk** to **<Installation directory>\HiCommandServer\database\interbase\HiCommand_backup.gbk.** Refer to section 2.3.2 for instructions.
6. Start the InterBase® Server service from the Services panel.
7. Select **Start → Programs → HiCommand → Restore Database**.
8. Start the services in the following order:
   - InterBase® server
   - InterServer®
   - HiCommand™ Suite single sign-on service
   - HiCommand™ Suite common web service
   - HiCommand™ server.

2.5.4 HiCommand™ Suite Single Sign-On Service or HiCommand™ Suite Common Web Service Fail to Start

If the single sign-on or common web service fail to start, possible causes include the following:

- The port number might already be in use. If this is the case, change the port number.
- A resource shortage occurred. If this is the case, either increase the swap area or increase the amount of installed memory.

If the single sign-on service or common web service still do not start, use the **hcmdsras** command to collect maintenance information, and then contact Customer Support. For details on the **hcmdsras** command, see section 6.2.

2.5.5 HiCommand™ Suite Single Sign-On Service or HiCommand™ Suite Common Web Service Do Not Appear in the Services Panel

If either the single sign-on service or the common web service does not appear in the Services panel, the installation of the common component may have failed.

1. Check the install log, which is in the installation directory. If you find the error, follow the instructions in the error message.
2. If problems still exist, uninstall Device Manager (see section 2.7.1 for instructions), reboot the system, and reinstall Device Manager (refer to section 2.2 for instructions).

2.5.6 HiCommand™ Server Fails to Start

If the HiCommand™ server fails to start, the installation may have failed. Examine the trace log at **<installation directory>\HiCommandServer\log\HDvMtrace*.log**, review the error codes, and follow the recommended action. For more information about error codes, see *HiCommand™ Device Manager Error Codes* (MK-92HC016).

2.5.7 HiCommand™ Server Does Not Appear in the Services Panel

If the HiCommand™ server does not appear in the Services panel, the installation may have failed. Uninstall Device Manager (see section 2.6.1), and reboot the system. Verify that all of the requirements have been met (refer to section 2.1.1), then reinstall Device Manager (refer to section 2.2 for instructions).

2.5.8 Other Problems

If you are experiencing other types of problems, the installation of Device Manager may have failed. Uninstall Device Manager (see section 2.7.1 for instructions), and reboot the system. Verify that all of the requirements have been met (refer to section 2.1.1), then reinstall Device Manager (refer to section 2.2 for instructions).

2.6 Upgrading or Reinstalling the Device Manager Server

If you want to install a newer version of the Device Manager server, install it over the existing version. Device Manager automatically updates your data and configurations to work with the latest version.

Warning: If you are upgrading from Device Manager version 2.2 or earlier to Device Manager 2.3 or later, the directory structure has changed. You need to delete the earlier version (see section 2.7 for instructions), and then do a clean install of the later version.

Warning: Do not cancel in the middle of an upgrade or reinstallation, because you could corrupt the files.

If you want to complete a manual backup before reinstalling the server, copy the following files and directories to a new location (e.g., **c:\Program Files\HiCommand\Device Manager\HiCommandBackup**):

- **<installation directory>\HiCommandServer\config*.properties**
- **<installation directory>\HiCommandServer\database\interbase\HiCommand.gdb**
- **<installation directory>\HiCommandServer\logs**
  (optional, if you want to back up the log files as well)
- **<installation directory>\HiCommandCLI\HiCommandCLI.properties**
- **<installation directory>\SupportTools\CollectTool\TIA.properties**

To upgrade or reinstall the Device Manager Server:

1. Log on to the system as an administrator.
2. Stop any other HiCommand™ Suite software that is running.
3. Stop the single sign-on service and the common web service by using one of the following methods:
   - Open the Windows® Services panel and stop the Device Manager server (refer to Figure 2.2 for the Windows® 2000 Services panel, and Figure 2.3 for the Windows NT® Services panel).
     Note: To access the Services panel in Windows® 2000, select **Start → Settings → Control Panel → Administrative Tools\Services**. In Windows NT®, select **Start → Settings → Control Panel → Services**.

   OR

   - Select **Start → Programs → HiCommand → Stop HiCommand**.

Instructions continue on the following page.

4. For version upgrades, Device Manager automatically upgrades old data files to the latest version.
5. ***Note***: If the HiCommand Suite is running panel (refer to Figure 2.5) is displayed, you must stop all of the HiCommand™ Suite Software before installation.
6. ***Warning***: You cannot downgrade Device Manager or upgrade to Device Manager LE. If you are attempting to do so, the installer will display the HiCommand Device Manager Upgrade Error panel (see Figure 2.41 and Figure 2.42, respectively) and exit.
7. Insert the Device Manager server CD to reinstall or upgrade the Device Manager server. The HiCommand Device Manager Already Installed panel (see Figure 2.43) displays.
8. Select **Next** to continue.
9. The rest of the installation should proceed as a normal installation. See section 2.2 for further details.

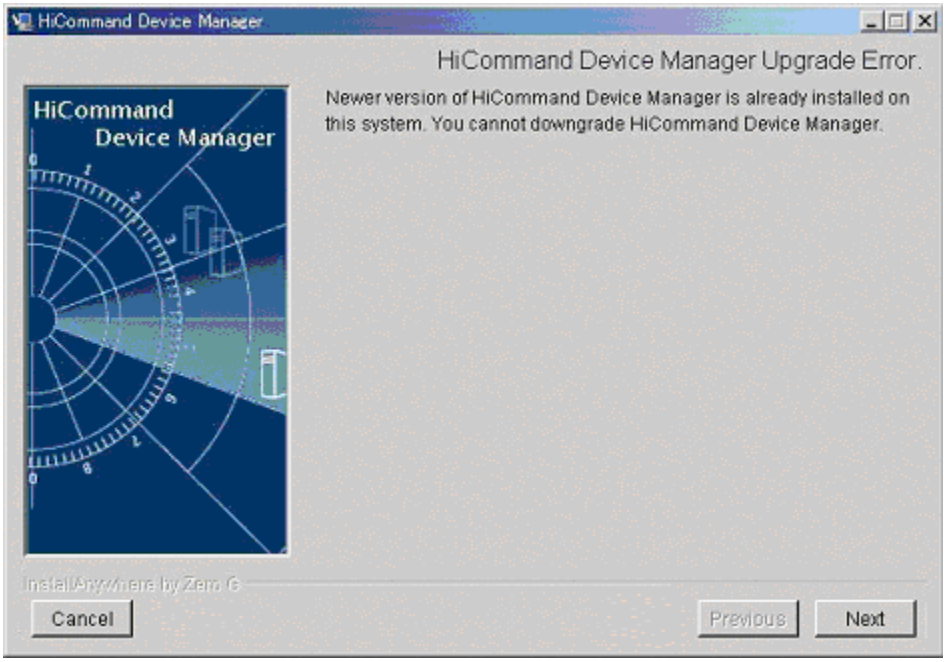

Figure 2.41 HiCommand Device Manager Upgrade Error Panel (Attempt to Downgrade)

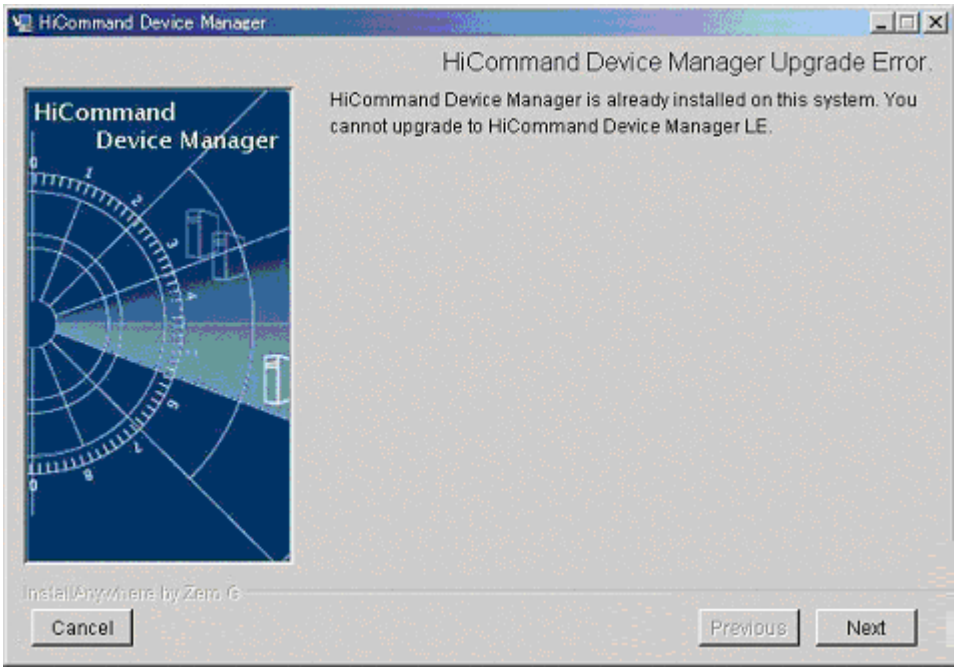

Figure 2.42 HiCommand Device Manager Upgrade Error Panel (Cannot Upgrade to LE)

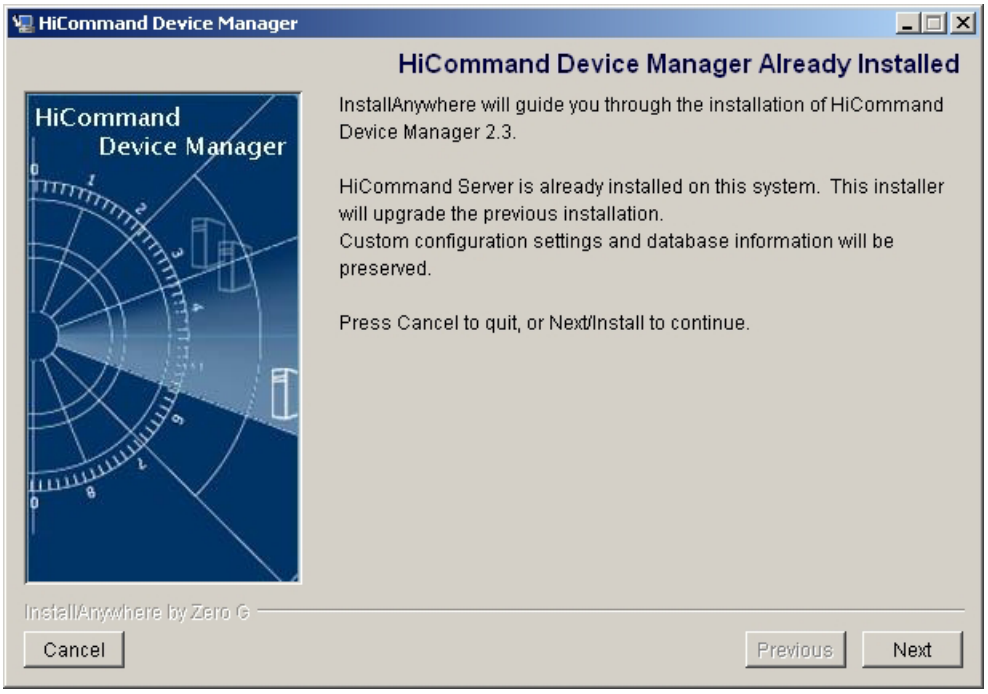

Figure 2.43 HiCommand Device Manager Already Installed Panel

2.7 Uninstalling Device Manager Components

2.7.1 Uninstalling the Device Manager Server

Warning: Unless you are experiencing problems and need to redo a complete installation, you should not uninstall Device Manager. This causes a loss of any defined configuration, including users, storage subsystems under management, and logical storage. If you are merely upgrading to a later version, refer to section 2.5 for instructions on upgrading or reinstalling the Device Manager server.

Warning: When you are using the single sign-on function, and Device Manager server and Tuning Manager are installed on the same machine, the single sign-on function is deleted when Device Manager is uninstalled.

If no other program is using the common component, it will be uninstalled during the uninstallation of Device Manager. If another program is using the common component, it will remain installed. You cannot uninstall the common component by itself.

When you uninstall Device Manager, the directories or files under **<installation directory>** will be deleted. The default installation directory has changed. For versions 2.2 and earlier, Device Manager was installed in **c:\program files\HiCommand**. For versions 2.3 and later, Device Manager will be installed in **c:\program files\HiCommand\Device Manager**, and the common component will be installed in **c:\program files\Hitachi\hntrlib2.** The files will be deleted, but the directory itself will remain after the uninstallation. If the directory is not necessary you can delete it manually.

Even though InstallAnywhere™ indicates that it uninstalls all components that were installed by that program, only the Device Manager server will actually be uninstalled. If you want to uninstall InterBase® or InterClient® software, you must do so separately (see sections 2.7.2 and 2.7.3).

To uninstall the HiCommand™ Device Manager:

1. If you have set the application startup information to the common component repository, you must remove this information. See section 4.3.3 for instructions.
2. If any other HiCommand™ Suite software is running, please stop that process.
3. Stop the single sign-on service and common web service. See section 4.2.2 for instructions.
4. Stop any other HiCommand™ Suite software that is running.
5. Stop the single sign-on service and the common web service from the Windows® Services Panel (see Figure 2.2 for the Windows® 2000 Services panel, and Figure 2.3 for the Windows NT® Services panel).
   Note: To access the Services panel in Windows® 2000, select **Start → Settings → Control Panel → Administrative Tools\Services**. In Windows NT®, select **Start → Settings → Control Panel → Services**. On the Services panel, select **HiCommand Server → Stop**.
6. Stop the Device Manager server by selecting **Start →Programs →HiCommand → Stop HiCommand**.
7. Use the Windows® **Add/Remove Programs** utility to access the InstallAnywhere™ Uninstaller panel (see Figure 2.44). Select **Uninstall** to continue. The Uninstall Complete panel (see Figure 2.45) displays.
8. Select **Quit** to exit. If the Windows® operating system indicates that certain files were not uninstalled, you must manually delete them.

Figure 2.44 InstallAnywhere™ Uninstaller Panel

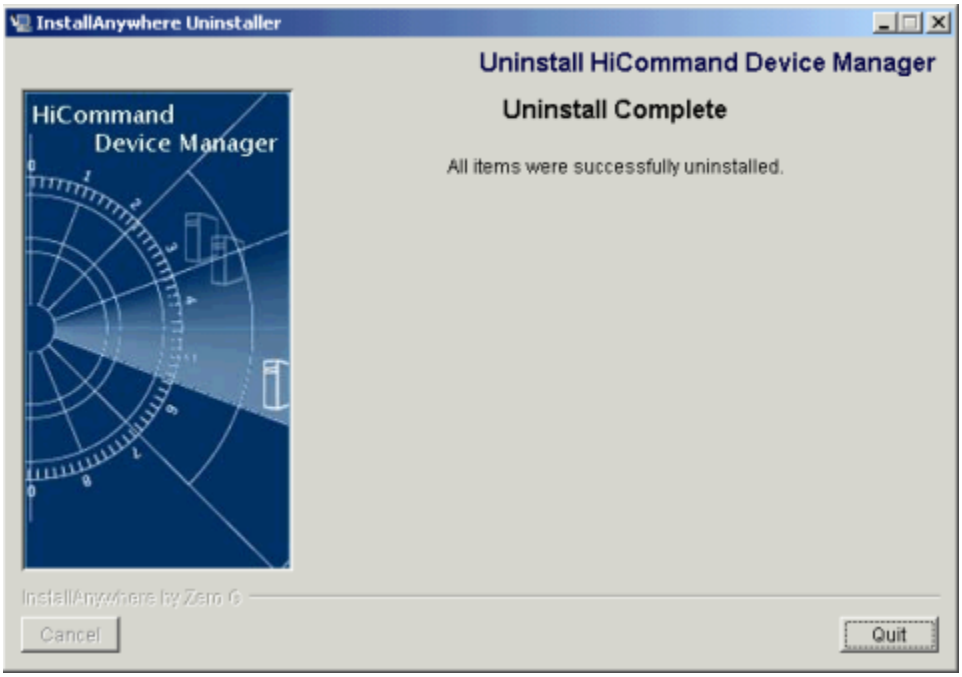

Figure 2.45 Uninstall Complete Panel

2.7.2 Uninstalling InterBase® Server and Client

Warning: Do not uninstall InterBase® Server and Client if another HiCommand™ Suite product is used on the same machine.

To uninstall InterBase® Server and Client:

1. Before you uninstall InterBase® Server, stop the service by accessing the Windows® Services Panel (refer to Figure 2.2 for the Windows® 2000 Services panel, and Figure 2.3 for the Windows NT® Services panel). On the Services panel, select **HiCommand Server → Stop,** then select **InterBase Guardian → Stop**.
   Note: To access the Services panel in Windows® 2000, select **Start → Settings → Control Panel → Administrative Tools\Services**. In Windows NT®, select **Start → Settings → Control Panel → Services**.
2. ***Note***: If you are using Windows® 2000, the **Microsoft Management Console** panel displays. Select **OK** to continue.
3. Use the Windows® **Add/Remove Programs** utility to remove the entries for the InterBase® software (the exact details are determined by your existing configuration).
4. The InterBase® uninstallation program may leave a directory structure on your system. The default location is **c:\Program Files\Borland\InterBase.**
5. Reboot the system.

Note: If you are also going to uninstall the InterClient® software, the reboot can wait until after InterClient® is uninstalled.

Note: If the uninstallation fails, retry the operation. If it continues to fail, restart the machine, reinstall InterBase® Server, and then uninstall InterBase® again. To install InterBase® Server, unzip the zip file in **<installation directory>\InterBaseInstallers\InterBaseServer**, then follow the instructions in this section.

Figure 2.46 Microsoft Error Message 997 (Can't Stop InterBase Guardian)

2.7.3 Uninstalling InterClient® Software

Warning: Do not uninstall InterClient® if another HiCommand™ Suite product is used on the same machine.

To uninstall the InterClient® software:

1. Before you uninstall InterClient®, stop the service by accessing the Windows® Services Panel (refer to Figure 2.2 for the Windows® 2000 Services panel, and Figure 2.3 for the Windows NT® Services panel). On the Services panel, select **Interserver→ Stop.**
   Note: To access the Services panel in Windows® 2000, select **Start → Settings → Control Panel → Administrative Tools → Services**. In Windows NT®, select **Start → Settings → Control Panel → Services**.
2. Use the Windows® Add/Remove Programs utility to remove the entries for the InterClient® software (the exact details are determined by your existing configuration).
3. The InterClient Uninstall Complete panel (see Figure 2.47) displays when you are finished.
   Note: If the InterClient® software is uninstalled, you must restart before you can reinstall.
4. ***Note***: If the uninstallation fails, retry the operation. If it continues to fail, restart the machine reinstall and then uninstall InterClient®. To reinstall InterClient®, execute the **IC20001winJRE.exe** file, which is located in **<installation directory>\InterBaseInstallers\InterClient**.

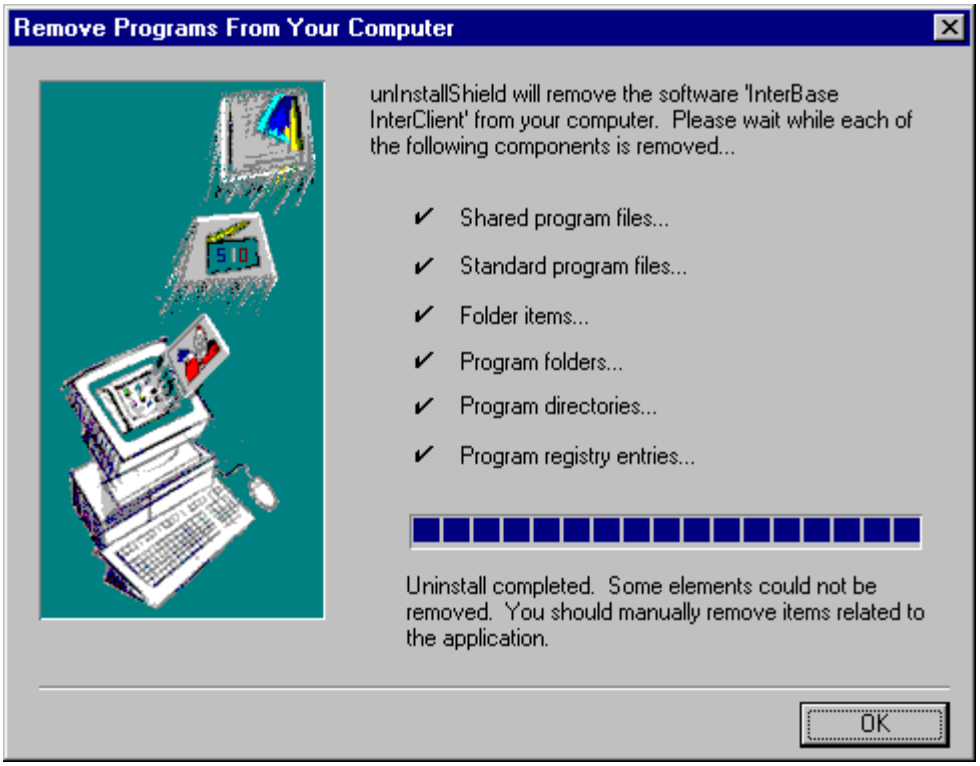

Figure 2.47 InterClient Uninstall Complete Panel

2.8 Backing Up and Restoring the Database

2.8.1 Backing Up the Database

You do not need to stop the Device Manager server before backing up the database. In Windows® systems you can back up the database using *one* of the following three methods:

To back up using the menu options:

- Select **Start → Programs → HiCommand → Back Up Database**.

OR

To back up using Windows® Explorer:

- Copy the **HiCommand.gdb** file to another location. The default location of the file is:
  C:\Program Files\HiCommand\Device Manager\HiCommandServer\database\interbase\HICOMMAND.GDB

OR

To back up the database using the database batch process:

1. At the command prompt, change directories to the Device Manager installation directory. The default installation directory has changed. For versions 2.2 and earlier, Device Manager was installed in **c:\program files\HiCommand**. For versions 2.3 and later, Device Manager will be installed in **c:\program files\HiCommand\Device Manager**, and the common component will be installed in **c:\program files\Hitachi\hntrlib2.** Select **Next** to continue**.** From the installation directory on Windows® systems, enter (including the quotation marks):
   database.bat backup "c:\Program files\backups\HiCommandBackup.gbk"

   The parameter within the quotation marks is the complete path and file name of the backup file to be created. If that parameter is not provided, **HiCommand_backup.gbk** is created in the same directory as the database.

2.8.2 Restoring the Database

Device Manager is installed with a backup of an empty database named **HiCommand.gbk**, which is located in the same directory as the database. Once the server has been stopped you can either restore a database that was previously backed up or you can restore an empty database. Restoring an empty database deletes any current configuration information.

Important: You must stop the Device Manager server before you restore the database.

2.8.3 Restoring an Existing Database

To restore an existing database using the database script:

1. Stop the Device Manager server by *one* of the following methods:
   - Access the Windows® Services Panel (see Figure 2.2 for the Windows® 2000 Services panel, and Figure 2.3 for the Windows NT® Services panel). On the Services panel, select **HiCommand Server → Stop.**
     Note: To access the Services panel in Windows® 2000, select **Start → Settings → Control Panel → Administrative Tools\Services**. In Windows NT®, select **Start → Settings → Control Panel → Services**.

     OR
   - Select **Start → Programs → HiCommand → Stop HiCommand**.
2. Invoke the database script with restore as the first parameter, and the absolute path of the backup file as the optional second parameter. If no backup file is specified, Device Manager searches the database directory for the file **HiCommand backup.gbk**. For example, from the installation directory on Windows® systems, enter (including the quotation marks):
   database.bat restore "c:\Program Files\backups\HiCommandBackup.gbk"
3. Restart the Device Manager server by *one* of the following methods:
   - Open the **Services** panel and restart the Device Manager server.

     OR
   - Select **Start → Programs → HiCommand → Start HiCommand**.

To restore an existing database using the menu:

1. Stop the Device Manager server by *one* of the following methods:
   - Open the Windows® Services panel (see Figure 2.2 for the Windows® 2000 Services panel, and Figure 2.3 for the Windows NT® Services panel). On the Services panel, select **HiCommand Server → Stop.**
     Note: To access the Services panel in Windows® 2000, select **Start → Settings → Control Panel → Administrative Tools\Services**. In Windows NT®, select **Start → Settings → Control Panel → Services** and stop the Device Manager server.

     OR
   - Select **Start → Programs → HiCommand → Stop HiCommand**.
2. Select **Start → Programs → HiCommand → Restore Database**. This restores the data from the default backup database in the database directory.
3. Restart the Device Manager server by *one* of the following methods:
   - Open the **Services** panel and restart the Device Manager server.

     OR
   - Select **Start → Programs → HiCommand → Start HiCommand**.

2.8.4 Restoring an Empty Database

Restoring an empty database is another way to initialize the configuration of the database.

To restore an empty database using the database script:

1. Stop the Device Manager server using *one* of the following methods:
   - From the Windows Services panel, select **HiCommand Server → Stop.**
     (Refer to Figure 2.2 for the Windows® 2000 Services panel, and Figure 2.3 for the Windows NT® Services panel).
     Note: To access the Services panel in Windows® 2000, select **Start → Settings → Control Panel → Administrative Tools\Services**. In Windows NT®, select **Start → Settings → Control Panel → Services**.

     OR
   - Select **Start → Programs → HiCommand → Stop HiCommand.**
2. Enter the following command to invoke the database script:
   database.bat restore
3. Restart the Device Manager server using *one* of the following methods:
   - Open the Services panel and restart the Device Manager server.

     OR
   - **Select Start → Programs → HiCommand → Start HiCommand**.

Chapter 3 Solaris™ Systems Installation

3.1 Overview

- Solaris™ system and media requirements (see section 3.1.1).
- Installing Device Manager (see section 3.2).
- Installing JRE™, InterBase® and InterClient® (see section 3.3).
- Verifying Device Manager and common component installation (see section 3.4).
- Reinstalling Device Manager (see section 3.5).
- Uninstalling Device Manager components (see section 3.6).
- Backing up and restoring the database (see section 3.7).

3.1.1 Solaris™ System and Media Requirements

The following are the system requirements for Sun® Solaris™:

- Workstation: SPARC™
- Operating System: Sun® Solaris™ 8 or 9. If you haven't already done so, download certain operating system patches, as follows:
  - Open your browser. In the URL field, enter:

    http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/patch-access
  - Locate the section entitled **Recommended Solaris™ Patch Clusters**.
  - Select either **Sun Solaris 8** or **Sun Solaris 9**
  - On the right side of the screen, select **Download FTP -> Go**.
  - Save the file **8 recommended.zip** or **9 recommended.zip** to your local disk.
  - Enter one of the following commands to unzip the file:

    unzip 8 recommended.zip

    unzip 9 recommended.zip

    This creates a directory with the same name but without the .**zip** file extension.
  - Change to the **recommended** directory and enter the following command to run the script:

    install_cluster
- Processor: 360 MHz or better.
- Memory (RAM): 256 MB or more.
- Available hard drive space: 4 GB or better.

Requirements continue on the following page.

- Monitor: VGA with 256 colors or better.
- CD-ROM drive.
- Static IP address (used to test the LAN connections and allow access to the Device Manager server).
- 9900V Series, 9900, 9500V, 9200 and/or T3 subsystem and Device Manager server residing on the same network.
- Minimum microcode/firmware requirements:
  - 9900 V Series microcode level: 21-01-50/00
  - 9900 microcode level: 01-13-19 if not using CVS (Virtual LUN) or LUSE functions, 01-15-39-00/05 if using CVS (Virtual LUN) or LUSE functions
  - 9500 V microcode level: 0651
  - 9200 microcode level: 0559
  - T3 firmware revisions: 1.1.7, 2.0.0
- Root access to install Device Manager.
- DNS/WINS/NIS awareness (for example, using the **NBSTAT** command to verify DNS awareness of the host if you want to use a registered hostname rather than the IP address in the URL).
- The platform running Device Manager has no other applications using the standard SNMP ports (161 and 162).

  Note: CA Unicenter TNG® and HP CommandView® systems can use these ports, so make sure to verify that these systems are not active.
- gunzip utility (Sun package SUNWgzip).
- unzip utility (Sun package SUNWzip).
- LAN cables, connections to the subsystem, and 10/100 Ethernet LAN card.

Warning: The 9900 and 9900V have a public LAN and a private LAN. Device Manager uses the public LAN to communicate with the SVP about the array and configuration changes. Do not *under any circumstances* attach the private LAN to an external network, because this can cause serious problems on the array. Figure 3.1 illustrates an incorrect LAN connection.

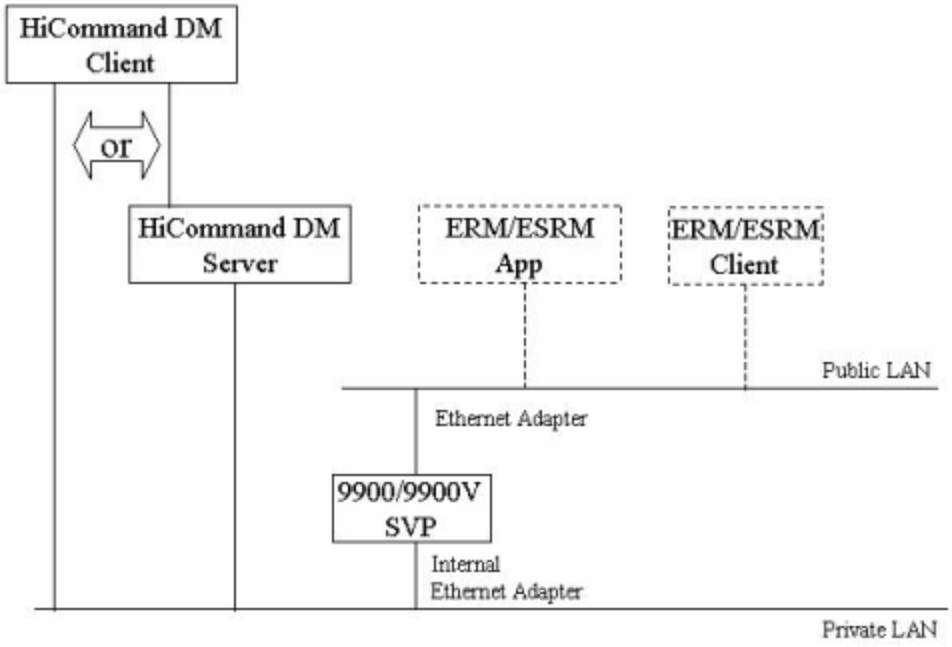

Figure 3.1 Incorrect 9900 and 9900V LAN Connection

3.1.2 Installation Overview

The Device Manager server installation CD includes the following applications:

- Java2™ JRE™ software: version 1.3.1_03
- InterBase® software: version 6.0.1 Server and Client
- InterClient® software: 2.0 JDBC Driver for InterBase®
- Device Manager Server software
- HiCommand™ Suite common component software.

Once these products are installed, the Device Manager client software can be downloaded, installed and updated on the client systems from the Device Manager server. Installation requires root access, so you must be logged in as **root** to run the installation. You can use the **su** command to assume root privileges to run the installation.

3.2 Installing the Device Manager Server and Common Component

Warning: If you are upgrading from Device Manager version 2.2 or earlier to Device Manager 2.3 or later, the directory structure has changed. You need to delete the earlier version (see section 3.6 for instructions), and then do a clean install of the later version.

Note: You will need about 400 MB of temporary disk capacity to install the Device Manager server.

1. Log onto the Solaris™ system as root.
2. If any other HiCommand™ Suite software is running, please stop that process. Then, stop the single sign-on service and common web service (see section 4.2.2 for instructions).
3. Insert the Device Manager CD-ROM. There are two files on the CD-ROM: **HDVM_02-30_sol.tar.gz** and **install.sh**. Execute **install.sh**.

   Warning: Do not use **Ctrl + c** to stop in the middle of the installation. If you stop the installation, run the **#pkginfo HDVM** command, and when the Device Manager information is displayed, uninstall (See section 3.6.3 for instructions) and then re-install the Device Manager server.

   Note: If lower (unsupported) versions of InterBase® or InterClient® software are found, the installer ends. Uninstall the lower version, and re-run the installer. For instructions on uninstalling these products, see section 3.6.1 for InterBase®, or section 3.6.2 for InterClient®.
4. The Device Manager is displayed as an available package. Press **Enter** to continue.

5. The HiCommand™ Device Manager Server License Agreement panel displays. Select **Yes** to each panel of the license agreement to continue.
6. The Device Manager package confirmation message contains scripts that are executed with super-user permission during the process of installing. Select **Yes** to start installation. Device Manager is installed in **/opt/HiCommand/**.
7. ***Note***: If the installer ends with an error, follow the instructions on the error message and reinstall Device Manager. If the command **pkginfo HDVM** displays the package information, first uninstall and then reinstall Device Manager. See section 3.6.3 for instructions on uninstalling the Device Manager server.

3.3 Installing Java2™ Java Runtime Environment™, InterBase® and InterClient®

3.3.1 Installing Java2™ Java Runtime Environment™

- If the JRE™ **License Agreement** displays, enter **yes** to continue. The Java2™ JRE™ software is installed in the **/opt/HiCommand/jre/** directory.
- If JRE™ software version 1.3.1_03 is found installed in the **/usr/j2re1_3_1_03/**, **/usr/java/jre/** or **/opt/HiCommand/jre/** directory, the JRE™ software installation is skipped automatically.

3.3.2 Installing InterBase® and InterClient®

The InterBase® database server used by Device Manager periodically creates and uses temporary files. The InterBase® software creates these temporary files in a directory specified by the InterBase® configuration file or by environment variables.

- A specified temporary directory must exist.
- The hard drive must have 20 MB of free space, or InterBase® will not run.
- If no entry is found in the configuration file, and the **INTERBASE_TMP** environment variable does not exist, the InterBase® software uses the directory pointed to by the system's **TMP** environment variable. You can use the default directory (**/tmp**), or you can specify other directories.
- You can also specify that the InterBase® software should use directories other than the default hard-coded directories.
  - To specify another directory, edit the InterBase® configuration file that is located in the InterBase® directory. On Solaris™ systems, the file is named **isc_config**.
  - Create an entry in this file in the following format:

    TMP_DIRECTORY size "pathname"
- You can also create the environment variable **INTERBASE_TMP** which points to the directory you would like the InterBase® software to use for temporary files.

Device Manager requires InterBase® Server 6.0.1, and InterClient® 2.0 JDBC Driver for InterBase®. If you have already installed these products at the specified version levels or higher, the installation process automatically skips this part of the installation.

To install InterBase® and InterClient® and complete the installation process:

If a lower version of InterBase® Server or InterClient® software is already installed the installer halts. Uninstall the lower version (see section 3.6.1 for InterBase®, or section 3.6.2 for InterClient®), and re-run the installer.

1. A confirmation of the InterBase® package message displays contains scripts that are executed with super-user permission during the installation process. Enter **yes** to begin.
2. The InterClient® message displays. Read it**,** and press **Enter** to continue.
3. Enter the installation directory, company name and your name to start the installation process. If the selected directory does not exist, a message will be displayed. Press **Enter** to continue
4. A message displays that the installation completed successfully. A successful installation of Device Manager automatically restarts the server.

3.3.3 Installing the HiCommand™ Suite Common Component

1. The common component is displayed as an available package, called **HiCommand Base**. Press **Enter** to continue.
   Note: If some of the common component files have already been installed, a message will display. Type **y** to continue.
2. The common component package confirmation message contains scripts that are executed with super-user permission during the installation process. Select **Yes** to start the installation. The common component is installed in **/opt/HiCommand/Base.**

3.4 Verifying Installation

3.4.1 Verifying Device Manager Installation

1. Enter the following command to verify that Device Manager is running:
   /opt/HiCommand/hicommand.sh status
2. This command outputs several lines of text. The first line of text should read:
   HiCommand process ID: xxxx
   (xxxx is a number representing the process ID for the Device Manager daemon)
3. If this is what you see, the installation was successful and Device Manager is up and running normally. However, if you see **"HiCommand is not running"**, this indicates a problem.
4. First verify that InterBase® database is running by typing the command:
   /opt/HiCommand/database.sh status
5. If the InterBase® database is running, you see a message similar to the following:
   InterBase Server is running with pid=14426
6. If the InterBase® database is not running, you see the following message:
   InterBase Server is not running
7. If the InterBase® database is not running, you can attempt to manually start the InterBase® database by running the script with the **start** parameter:
   /opt/HiCommand/database.sh start
8. After running the database script, run the script again with the **status** parameter to verify that the database has started:
   /opt/HiCommand/database.sh status
9. If the database is running, manually start the Device Manager by executing the following command:
   /opt/HiCommand/hicommand.sh start

3.4.2 Verifying HiCommand™ Suite Common Component Installation

1. Verify the common web server process by executing the following command:
   /etc/init.d/hicommand-CWS check
2. If the common web server is not operating, start it by executing the following command:
   /etc/init.d/hicommand-CWS start
3. Verify the single sign-on process by executing the following command:
   /etc/init.d/hicommand-SSOS check
4. If the single sign-on process is not running, start it by executing the following command:
   /etc/init.d/hicommand-SSOS start
5. If one or both of these processes are still not running, please check the install log, which is located in Device Manager Installation directory.
6. If you find the error message in the log follow the instructions.
7. If the problem still remains, uninstall Device Manager (see section 3.6.3 for instructions), reboot the system, and reinstall Device Manager (refer to section 3.2 for instructions).

3.5 Upgrading or Reinstalling the HiCommand™ Device Manager Server Software

To upgrade the Device Manager, install the newer version of Device Manager over the existing version. Device Manager automatically updates your configuration and saves the data to the latest format.

Warning: If you are upgrading from Device Manager version 2.2 or earlier to Device Manager 2.3 or later, the directory structure has changed. You need to delete the earlier version (see section 3.6 for instructions), and then do a clean install of the later version.

Warning: Do not use **Ctrl+c** to cancel in the middle of an upgrade or reinstallation because you could corrupt the files.

1. Log onto the Solaris™ system as root.
2. If any other HiCommand™ Suite software is running, please stop that process. Then, stop the single sign-on service and common web service (see section 4.2.2 for instructions).
3. Stop the Device Manager server by entering the **stop** command from a terminal window: **<installation directory>/hicommand.sh stop**
4. Verify that Device Manager has been stopped by entering the **status** command: **<installation directory>/hicommand.sh status**
5. The output of this command should display several lines of text. The second line of the output should display the text: **HiCommand is not running**. If a process ID displays, Device Manager is still running. Return to step 3 to stop the server.
6. For version upgrades, Device Manager automatically upgrades old data files to the latest version. However, if you want to do a manual backup, include the following files and directories:
   - **<installation directory>/HiCommandServer/config/*.properties**
   - **<installation directory>/HiCommandServer/database/interbase/HiCommand.gdb**
   - **<installation directory>/HiCommandServer/logs** (this is optional, if you want to back up the log files).
   - **<installation directory>/HiCommandCLI/HiCommandCLI.properties**
   - **<installation directory>/SupportTools/CollectTool/TIA.properties**

7. ***Note***: If the older version of Device Manager was not installed in **/opt/HiCommand/**, the upgrade of Device Manager will not execute properly. In this case, do a manual backup, uninstall the older version of Device Manager by executing the command **<installation directory>/UninstallerData/Uninstall_HiCommand -i console,** then reinstall the newer version of Device Manager (refer to section 3.2 for instructions).
8. Insert the Device Manager CD-ROM. There are two files on the CD-ROM: **HDVM_02-30_sol.tar.gz** and **install.sh**. Execute **install.sh**.
9. ***Note***: If lower (unsupported) versions of InterBase® or InterClient® software are detected, the installer ends. Uninstall the lower version, and re-run the installer. See section 3.6 for instructions on how to uninstall the InterBase® and InterClient® software.
10. The Device Manager is listed as an available package. Press **Enter** to continue.
11. A message displays that indicates that the installing files are in use by other package. Press **Enter** and answer **yes** to the verification prompts.
12. A message displays confirming the Device Manager package installation that contains scripts that are executed with super-user permission during the installation process. Answer **yes** to start the installation. Device Manager is installed in the **/opt/HiCommand/** directory.
13. The common component is displayed as an available package. Select **Enter** to continue.
14. If the common component is already installed, a message displays that indicates that the installing files are in use by other package. Press **Enter** and answer **Yes** to the verification prompts.
15. The common component package confirmation message contains scripts that are executed with super-user permission during the installation process. Answer **Yes** to start installation. common component is installed in **/opt/HiCommand/Base**.
16. If Java™ Runtime Environment (JRE)® (version 1.3.1_03) is not installed in **/usr/java/jre** or **/usr/j2re1_3_1_03**,
    it will be installed to **/usr/j2re1_3_1_03** to make a symbolic link from **/opt/HiCommand/jre**.
17. A successful installation of Device Manager automatically restarts the server. If you want to restore the data files, stop the server as directed above.
18. ***Note***: If the installer ends with an error, follow the instructions in the error message, then reinstall Device Manager.
19. If the command **pkginfo HDVM** displays the package information, first uninstall and then reinstall Device Manager (see section 3.6.3 for instructions). Be sure to create a backup before uninstalling Device Manager. See section 3.7 for instructions.

3.6 Uninstalling HiCommand™ Device Manager Components

3.6.1 Uninstalling InterBase® Software

Removing the InterBase® software is optional, unless you have an older version of the software that you need to upgrade in order to successfully install or re-install Device Manager.

Warning: If another HiCommand™ Suite product is being used on the same machine, do not uninstall InterBase®.

To uninstall InterBase®:

1. Log on as root or with root privileges. Enter the command:

   pkgrm IBCSN60

2. Answer **yes** to the verification prompts
3. Stop InterBase® by using one of the following methods:
   - If Device Manager is already installed:

     /opt/HiCommand/database.sh stop

   - If Device Manager is not already installed:

     kill –TERM 'ps –ef |grep ibserver|grep –v grep|awk' {print $2}''

4. ***Note***: A directory structure may have been left on disk as a result of the InterBase® installation log file that is generated. Enter the following command to safely remove this directory:
   rm -rf /opt/interbase

3.6.2 Uninstalling InterClient® Software

Warning: If another HiCommand™ Suite product is being used on the same machine, do not uninstall InterClient®.

To uninstall the InterClient® software:

1. Enter the following command:
   rm -rf /usr/interclient

   Note: If **/usr/interclient** is a link, then you also need to delete its linked directory.
2. Delete the entry for the InterServer® software in the **/etc/services file** (for example, interserver 3060/tcp).
3. Delete the entry for the InterServer® software in the **/etc/ometd/conf** file.
4. Either reboot the system, or force the **inetd** daemon to re-read its configuration file by entering the following command:
   kill -HUP 'ps -ef | grep initd | grep -v grep | awk '{print $2}''

3.6.3 Uninstalling the HiCommand™ Device Manager Server

Warning: Unless you are experiencing problems and need to redo a complete installation, you should not uninstall Device Manager. Uninstalling loses any previous configuration, such as users, storage subsystems, and logical storage groups. If you are merely upgrading to a later version, refer to section 3.5 for instructions on reinstalling the Device Manager server.

Warning: When you are using the single sign-on function and a Device Manager server and Tuning Manager are installed on the same machine, single sign-on service is deleted when Device Manager is uninstalled.

To uninstall the HiCommand™ Device Manager Server:

1. If you have set the application startup information to the common component repository, you have to remove this information (see section 4.3.3 for instructions).
2. If any other HiCommand™ Suite software is running, please stop that process. Then, stop the single sign-on service and common web service (see section 4.2.2 for instructions).
3. Stop the Device Manager server by entering the following command:
   /opt/HiCommand/hicommand.sh stop
4. Verify the status of the Device Manager by entering the following command:
   /opt/HiCommand/hicommand.sh status
5. The output of this command should display several lines of text. The second line of the output should display the text **HiCommand is not running**. If the output displays a process ID, Device Manager is still running. Return to step 3 to stop the server.
6. If Device Manager was successfully stopped, make note of the Device Manager installation directory displayed among the output of the command.
7. Enter the Device Manager package removal command:
   /opt/HiCommand/Uninstall/uninstall.sh
8. A message confirming that you want to delete Device Manager and the common component is displayed. Enter **y** to begin the uninstallation.

Note: Uninstalling Device Manager does not uninstall InterBase® and InterClient® software. Refer to section 3.6.1 for uninstallation instructions for InterBase®, or section 3.6.2 for instructions for InterClient®. ***Note***: Files can remain in the installation directory. Do not delete this directory if another HiCommand™ Suite product is being used.

Note: If no program is using the HiCommand™ Suite common component, the common component will be uninstalled during the uninstallation of the Device Manager server. If one or more other programs are using the common component, it will be uninstalled only when you uninstall the last program that is using it.

3.7 Backing Up and Restoring the Database

3.7.1 Backing Up the Database

1. Stop the Device Manager server. From a terminal window, enter the command:
   /opt/HiCommand/hicommand.sh stop
2. Verify that the Device Manager is not running. Enter the command:
   /opt/HiCommand/hicommand.sh status
3. The output of this command should display several lines of text. The second line of the output should display the text **"HiCommand is not running"**. If instead it displays a process ID, Device Manager is still running. Return to step 1 to stop the server.
4. To perform a database backup to a specified file, use the database script with the following parameters:
   database.sh backup <backup filename>
5. For example, from the installation directory, enter the command:
   database.sh backup /usr/backups/HiCommandBackup.gbk
6. If you don't specify a backup file, the backup file is created with the name **HiCommand_backup.gbk** and placed in the same directory as the database. The default location of the database is:
   /opt/HiCommand/HiCommandServer/database/interbase
7. Restart the Device Manager server by issuing the following command:
   <**installation directory>/hicommand.sh start**
8. This command outputs several lines of text. The first line of text should read:
   HiCommand process ID: xxxx. This indicates that Device Manager is up and running normally.
   ***Note*: xxxx** is a number representing the process ID for the Device Manager daemon.

3.7.2 Restoring the Database

Important: You must stop the Device Manager server before you restore the database from a backup. Refer to section 3.7.1 for instructions.

To restore the database:

1. Restore your Device Manager database from a backup database named **HiCommandBackup.gbk** located in the **/usr/backups** directory. Enter the command: **database.sh restore /usr/backups/HiCommandBackup.gbk**
2. ***Note***: If the location of the backup file is not specified, the server searches for this file in the same directory as the database. The default location of the database is: **/opt/HiCommand/HiCommandServer/database/interbase/**
3. Start the Device Manager server by entering the following command: **/opt/HiCommand/hicommand.sh start**
4. This command outputs several lines of text. The first line of text should read: **HiCommand™ process ID: xxxx.** This indicates that Device Manager is up and running normally.

   ***Note*: xxxx** is a number representing the process ID for the Device Manager daemon.

Chapter 4 HiCommand™ Suite Common Component

The HiCommand™ Suite common component provides features that are used by all HiCommand™ Suite products. Each HiCommand™ Suite product will bundle the common component. This chapter discusses the following functions:

- Installing and uninstalling the common component (see section 4.1)
- Starting and stopping the common component (see section 4.2)
- Enabling and disabling single sign-on (see section 4.3)
- Using the Integrated Log files (see section 4.4)

Table 4.1 describes the common component elements that are used by Device Manager.

Table 4.1 Device Manager Common Component Elements

Function	Description
Single sign-on	A customer who operates multiple HiCommand™ Suite products is not prompted to re-enter their user ID and password if that customer is using those products simultaneously. Single sign-on provides a unified user authentication mechanism.
Integrated logging information	Operation and other types of logs are concentrated by integrated logging information feature. Providing a common log repository allows all HiCommand™ Suite log files to be in the same file.

4.1 Installing and Uninstalling the HiCommand™ Suite Common Component

The common component must be installed or uninstalled as part of the installation or uninstallation of another HiCommand™ product, e.g., Device Manager or Tuning Manager. You cannot install or uninstall just the common component.

Whether the common component is either installed or upgraded during Device Manager installation is determined by following factors:

- If the common component is not previously installed on the system, the Device Manager installer will install it.
- If the previously installed common component version is older than or equal to the installing common component version, the Device Manager installer will upgrade the common component by overwriting the previous installation.
- If the previously installed common component version is newer than the installing common component version, the previously installed version is left intact.

Whether the common component is uninstalled or left in place during Device Manager uninstallation is determined by the following factors:

- If any other HiCommand™ Suite products are using the common component, the common component will not be uninstalled. The common component will not be uninstalled until the last HiCommand™ Suite product that uses common component is uninstalled.
- If the common component is being used by only by Device Manager, it will be uninstalled as part of the Device Manager uninstallation process.

4.2 Starting and Stopping the HiCommand™ Suite Common Component

4.2.1 Starting the Common Component

To start the common component in a Windows® environment:

1. Access the Windows® Services panel (refer to Figure 2.2 for the Windows® 2000 Services panel, and Figure 2.3 for the Windows NT® Services panel).
   Note: To access the Services panel in Windows® 2000, select **Start → Settings → Control Panel → Administrative Tools → Services**. In Windows NT®, select **Start → Settings → Control Panel → Services**.
2. Start the services in the following order:
   - Single sign-on service
   - Common web service

To start the common component in a Solaris™ environment:

1. Start the single sign-on process:
   /etc/init.d/hicommand-SSOS start
2. Start the common web service:
   /etc/init.d/hicommand-CWS start

4.2.2 Stopping the Common Component

To stop the common component in a Windows® environment:

1. Access the Windows® Services panel (refer to Figure 2.2 for the Windows® 2000 Services panel, and Figure 2.3 for the Windows NT® Services panel).
   Note: To access the Services panel in Windows® 2000, select **Start → Settings → Control Panel → Administrative Tools\Services**. In Windows NT®, select **Start → Settings → Control Panel → Services**.
2. Stop the services in the following order:
   - Single sign-on service
   - Common web service

To stop the common component in a Solaris™ environment:

1. Stop the single sign-on process:
   /etc/init.d/hicommand-SSOS stop
2. Verify that the single sign-on process has stopped:
   /etc/init.d/hicommand-SSOS check
3. Stop the common web service:
   /etc/init.d/hicommand-CWS stop
4. Verify that the common web service has stopped:
   /etc/init.d/hicommand-CWS check

4.3 Using Single Sign-On to Launch Other HiCommand™ Suite Products

4.3.1 Setting the Application Startup Information

Device Manager and Tuning Manager each have their own authentication functions and provide a GUI to enter your user ID and password. The integrated single sign-on mechanism is used in conjunction with the Device Manager and Tuning Manager application startup information, so that once you enter your User ID and password into one HiCommand™ product, you don't have to enter them again when you launch other HiCommand™ products.

Note: Device Manager version 2.3 is able to launch HiCommand™ Tuning Manager version 1.1 or later with the single sign-on mechanism.

Requirements for single sign-on:

- Tuning Manager version 1.1 or later
- Device Manager version 2.3 or later
- Device Manager and Tuning Manager installed on the same server.
- The application startup information set up in the common component repository.

Note: The URL for Device Manager must be accessible from other HiCommand™ products. If you configure a secure socket using Device Manager, the URL must specify the server name used to create the Keypair. For more information on configuring secure sockets, see *HiCommand™ Device Manager Server Security Guide* (MK-92HC003).

4.3.2 Enabling the Application Startup Information

To enable the common component single sign-on function, you need to execute the **hcmdsrep** command, which will set the URL of Device Manager and Tuning Manager in the common component repository.

The **hcmdsrep** command has the following features:

- The **hcmdsrep** command is located in **<common component installation directory>\Base**.
- The general format of the **hcmdsrep** command is:
  hcmdsrep /add /type /DeviceManager /url /user /pass
  - **url** indicates the Device Manager URL, which must be accessible from the other HiCommand™ Suite products.
  - **user** indicates the user name, which is generally the Device Manager system administrator.
  - **pass** indicates the password of the system administrator.
- The delimiter is **/** for Windows®, and **-** for Solaris™.
- An example of a **hcmdsrep** command is:

  hcmdsrep /add /type /DeviceManager /url http://172.16.17.39:2001 /user system /pass manager

4.3.3 Deleting the Application Startup Information

You will need to remove the Link and Launch information when you uninstall Device Manager. To disable Link and Launch, you must delete the Device Manager URL from the common component repository by executing the **hcmdsrep** command.
Note: For instructions on deleting the Tuning Manager URL from the common component repository see the Tuning Manager documentation.

Note: In order to disable Link and Launch, the following services must be running:

- Device Manager
- Single sign-on service
- Common web service

The **hscmdsrep** command has the following features:

- The **hcmdsrep** command is located in **<common component installation directory>\Base\bin**.
- HiCommand™ Device Manager, single sign-on service, and common web service must be running.
- An example of the **hcmdsrep** command for Windows® is:

  hcmdsrep /add /type DeviceManager /url DeviceManager-starting-URL/service/Launcher /user DeviceManager-system-administrator-user-ID /pass DeviceManager-system-administrator-password
- An example of a **hcmdsrep** command for Solaris™ is:

  hcmdsrep -add -type DeviceManager -url http://123.12.3.4:2001/service/Launcher -user user01 -pass user01

4.4 Integrated Logging

The common component provides common log files and a common library for log output for each program product in the HiCommand™ Suite. Device Manager uses this information to show the details for the log files.

Table 4.2 Integrated Log Output

Log Type	Log Name	Description	Location (Windows®)	Location (Solaris™)
Common trace log file	hntr2*.log	Integrated trace log information produced by the common component. The asterisk (*) in the file name indicates a file number	C:\Program Files\Hitachi\HNTRLib2\ spool	/var/opt/hitachi/HNTR Lib2/spool
Event log/syslog file	Eventlog	Windows® event log	Event viewer	N/A
	syslog	Solaris™ system log	N/A	Defined by /etc/syslog.co
Device Manager log file	version	Version information about the operating environment of the Device Manager server (Device Manager server, Java® VM, and operating system	<installation directory>/HiCommand Server/logs	
Device Manager log file	HDvMtrace*. log	Trace log information for the Device Manager server output by the common component. The asterisk (*) in the file name indicates a file number		

4.5 Ports Used By the HiCommand™ Suite Common Component

4.5.1 Device Manager Ports Used By The Common Component

Table 4.3 lists the ports used by the HiCommand™ Suite common component and Device Manager server.

Table 4.3 Ports Used by HiCommand™ Device Manager and Common Component

Component	Network Port	Description
Device Manager Server.	2001/tcp	Used for the Device Manager HTTP (web) server.
	snmp:161/udp	May be used for SNMP in the future. If another application is using this port, use caution when you upgrade Device Manager
	snmptrap:162/udp	Used for receiving SNMP traps from the subsystem(s).
	interserver:3060/tcp	Used for Device Manager internal communication.
	gds_db:3050/tcp	Used for Device Manager internal communication.
Device Manager Agent. ***Note***: For more information see HiCommand™ Device Manager Agent Installation Guide (MK-92HC019).	23011/tcp	Used for the Device Manager Agent HTTP (web) server.
	23012/tcp	Used for communication between Device Manager Agent's daemon process and the web server process.
	23013/tcp	Used for Device Manager Agent's daemon process (or service). Use the property file to change the port number.
HiCommand™ Suite common component.	23015/tcp	The Device Manager server uses this port to access non-SSL common web service. See section 4.5.2.1.
	23016/tcp	Web browsers use this port to access SSL common web service. See section 4.5.2.2.
	23017/tcp	The common web service uses this port to access single sign-on service through an AJP connection. See section 4.5.2.3
	23018/tcp	The single sign-on service uses this port to receive a stop request. See section 4.5.2.4
	23019/tcp	The common web service uses this port to access Tuning Manager through an AJP connection. See section 4.5.2.5.
	23020/tcp	Tuning Manager uses this port to receive a stop request. See section 4.5.2.6.
	23021/tcp	The common web service uses this port to access the Tuning Manager SOAP-API service through an AJP connection. See section 4.5.2.7.
	23022/tcp	Tuning Manager SOAP-API service uses this port to receive a stop request. See section 4.5.2.8.
	23023/tcp to 23034/tcp	Reserved

4.5.2 Changing Common Component Ports

4.5.2.1 23015/tcp (Used for Accessing Non-SSL HiCommand™ Suite Common Web Service)

To change the port used for accessing non-SSL common web service, you must change the port number written in the following file:

Windows®:

- The listen directive in **<installation directory >\Base\httpsd\conf\httpsd.conf**

Solaris™:

- The listen directive in **/opt/HiCommand/Base/httpsd/conf/httpsd.conf**

Note: If you want to execute the single sign-on function, you must also change the port number in the following file:

Windows®:

- **hsso.hostport** in **<installation directory>\Base\conf\hsso.conf**

Solaris™:

- **hsso.hostport** in **/opt/HiCommand/Base/conf/hsso.conf**

4.5.2.2 23016/tcp (Used For Accessing SSL HiCommand™ Suite Common Web Service)

To change the port used for accessing SSL common web service, you must change the port number written in the following files:

Windows®:

- **VirtualHost host-name:port-number** in **<installation directory>\Base\httpsd\conf\httpsd.conf**

  AND

- The **listen** directive writing the port number in **VirtualHost host-name:port-number**.

Solaris™:

- **VirtualHost host-name:port-number** in **/opt/HiCommand/Base/httpsd/conf/httpsd.conf**

  AND

- The **listen** directive writing the port number in **VirtualHost host-name:port number.**

4.5.2.3 23017/tcp (Used for HiCommand™ Suite Single Sign-On Service Through an AJP Connection)

To change the port used for the single sign-on service through an AJP connection, you must change the port number written in the following files:

Windows®:

- **worker.worker1.port** in **<installation directory>\Base\hwc\redirector\workers.properties**

  AND

- **webserver.connector.ajp13.port** in **<installation directory>\Base\hwc\containers\HiCommand\usrconf\usrconf.properties**

Solaris™:

- **worker.worker1.port** in **/opt/HiCommand/Base/hwc/redirector/workers.properties**

  AND

- **webserver.connector.ajp13.port** in **/opt/HiCommand/Base/hwc/containers/HiCommand/usrconf/usrconf.properties**

4.5.2.4 23018/tcp (Used for Stop Requests to HiCommand™ Suite Single Sign-On Service)

To change the port through which the single sign-on service receives a stop request, you must change the port number written in the following file:

Windows®:

- **webserver.shutdown.port** in **<installation directory>\Base\hwc\containers\HiCommand\usrconf\usrconf.properties**

Solaris™:

- **webserver.shutdown.port** in **/opt/HiCommand/Base/hwc/containers/HiCommand/usrconf/usrconf.properties**

4.5.2.5 23019/tcp (Used for Accessing Tuning Manager Through an AJP Connection)

To change the port used for Tuning Manager through an AJP connection, you must change the port number written in the following files.

Windows®:

- **worker.worker2.port** in **<installation directory>\Base\hwc\redirector\workers.properties**

  AND

- **webserver.connector.ajp13.port** in **<installation directory>\Base\hwc\containers\TuningManager\usrconf\usrconf.properties**

Solaris™:

- **worker.worker2.port** in **/opt/HiCommand/Base/hwc/redirector/workers.properties**

  AND

- **webserver.connector.ajp13.port** in **/opt/HiCommand/Base/hwc/containers/TuningManager/usrconf/usrconf.properties**

4.5.2.6 23020/tcp (Used for Tuning Manager Stop Requests)

To change the port through which Tuning Manager receives a stop request, you must change the port number written in the following file:

Windows®:

- **webserver.shutdown.port** in **<installation directory>\Base\hwc\containers\TuningManager\usrconf\usrconf.properties**

Solaris™:

- **webserver.shutdown.port** in **/opt/HiCommand/Base/hwc/containers/TuningManager/usrconf/usrconf.properties**

4.5.2.7 23021/tcp (Used For Tuning Manager SOAP-API Service Through AJP connection)

To change the port used for Tuning Manager SOAP-API service through AJP connection, you must change the port number written in the following files:

Windows®:

- **worker.worker3.port** in **<installation directory>\Base\hwc\redirector\workers.properties**

  AND

- **webserver.connector.ajp13.port** in **<installation directory>\Base\hwc\containers\TuningService\usrconf\usrconf.properties**

Solaris™:

- **worker.worker3.port** in **/opt/HiCommand/Base/hwc/redirector/workers.properties**

  AND

- **webserver.connector.ajp13.port** in **/opt/HiCommand/Base/hwc/containers/TuningService/usrconf/usrconf.properties**

4.5.2.8 23022/tcp (Used For Tuning Manager SOAP-API Stop Requests)

To change the port through which Tuning Manager SOAP-API service receives a stop request, you must change the port number written in the following file:

Windows®:

- **webserver.shutdown.port** in **<installation directory>\Base\hwc\containers\TuningService\usrconf\usrconf.properties**

Solaris™:

- **webserver.shutdown.port** in **/opt/HiCommand/Base/hwc/containers/TuningService/usrconf/usrconf.properties**

Chapter 5 HiCommand™ Device Manager Server Properties

5.1 Overview of the HiCommand™ Device Manager Server Properties

- **Server web configuration properties** (see section 5.2.1) include the IP address and port of the HTTP listener(s), the location of the server's document directory, and the name of the default index page. Server performance properties include the size of input/output buffers, various TCP/IP stack and socket settings, server file-cache parameters, and connection thread priorities.

  Warning: You should not undertake the task of optimizing these attributes unless you are an expert, since minor changes could severely impact the performance of the Device Manager server.

- **Server database properties** (see section 5.2.2) include DBMS parameters, such as drivers, logon ID and as debugging and optimization property settings for the Java™ Database Connectivity (JDBC) layer in the Device Manager server.

  Warning: You should not undertake the task of optimizing these attributes unless you are an expert, since minor changes could severely impact the performance of the Device Manager server.

- **Server logger properties** (see section 5.2.3) include directives that configure Device Manager server's logging module, including the names, locations and verbosity level of operational and error logging of the various log files.

- **Server dispatcher properties** (see section 5.2.4) include properties that allow the fine-tuning of various background processes (daemons) and the optimization of the thread-priority for service agents.

- **Server MIME properties** (see section 5.2.5) include the translation/lookup table for all Multipurpose Internet Mail Extensions (MIME) file types recognized by the Device Manager web server.

- **Client properties** (see section 5.2.6) include properties that configure the Device Manager Web Client.

- **Server security properties** include whether secure-socket encryption is being utilized, the location and passwords for the Server Certificate TrustStore, and a list of permitted client IP addresses. This group also contains a number of properties that support the hardening of the Device Manager server against certain kinds of denial-of-service attacks. Please refer to the *HiCommand™ Device Manager Server Security Guide* (MK-91HC003) for a description of security properties.

There may also be certain debugging properties defined in the server configuration file. Because these attributes are intended only for use in problem-solving situations that require highly detailed diagnostic information, they may be absent from production releases of the configuration files. These files are in Java™ property file format, and *except for the security properties file* can be modified using any text editor. Each property directive consists of a name-value pair separated by the equal sign (for example, foo.bar=12345)**.** The appropriate end-of-line terminator, as defined by the operating system, delineates individual properties.

The configuration files for the Device Manager server are located in a **config** folder under the installation directory.

The default directory for the Windows® configuration files is:

C:\Program Files\HiCommand\Device Manager\HiCommandServer\config

The directory for Solaris™ configuration files is:

/opt/HiCommand/HiCommandServer/config

Comments in Device Manager property files are tagged using the "#" character at the start of a line. Literals (text strings or numeric values) do not need to be quoted. Boolean values can be either **true** or **false** (case-insensitive). Any other setting (for example, **yes**) is interpreted as **false**.

The backslash is a reserved character in Java™ property files, and is used for escaping various control characters such as tabs, line-feeds, etc. On Windows® platforms absolute pathnames typically contain backslash characters, and must be backslash-escaped, for example, the file pathname **c:\HiCommand\docroot\foo.bar** should be entered as **c:\\HiCommand\\docroot\\foo.bar**. There is generally no need to backslash-escape any other characters in the property directives.

***Warning*:** Use extreme caution when you are modifying the configuration properties, because you can cause the server to fail or to function incorrectly. Don't attempt modification of server properties unless you have sufficient expertise to understand the potential consequences.

Table 5.1 summarizes the various Device Manager property files. Please refer to the *HiCommand™ Device Manager Server Security Guide* (MK-91HC003) for a description of security properties.

Table 5.1 Summary of Device Manager Property Files (continues on the following pages)

Property	Description	Location
server.http.host	Designates either the host name or the dotted-decimal IP address for the Device Manager web server.	Section 5.2.1.1
server.http.socket.bindaddress	Specifies the interface or interfaces where Device Manager should listen.	Section 5.2.1.2
server.http.port	Assigns the port used for the Device Manager HTTP server.	Section 5.2.1.3
server.https.port	Assigns the port used for the Device Manager secure HTTP server.	Section 5.2.1.4
server.http.default	Sets the name of the default index page for the Device Manager web server.	Section 5.2.1.5
server.http.user	Sets the user identity for the UNIX® user who owns the Device Manager server process.	Section 5.2.1.6
server.http.group	Sets the user group for the UNIX® user who owns the Device Manager server process.	Section 5.2.1.7
server.http.chroot	Flags whether to invoke a **chroot** (change root) command into the Device Manager server's installation directory.	Section 5.2.1.8
server.http.request.timeout	Sets the read-blocking timeout of the HTTP socket connection.	Section 5.2.1.9
server.http.connection.priority	Sets the priority for all client-connection threads spawned by HTTP requests made against the Device Manager server.	Section 5.2.1.10
server.http.connection.bufSize	Sets the size (in bytes) for all of the server's input/output (I/O) buffers.	Section 5.2.1.11
server.http.socket.backlog	Assigns the maximum queue length for incoming connection indications.	Section 5.2.1.12
server.http.socket.maxThreads	Sets the maximum number of concurrent connections accepted by the Device Manager server.	Section 5.2.1.13
server.http.socket.linger	Toggles whether the **SO_LINGER** socket attribute is enabled for client connections with the Device Manager server.	Section 5.2.1.14
server.http.socket.noDelay	Toggles whether the **TCP_NODELAY** socket attribute is enabled for connections to the Device Manager server.	Section 5.2.1.15
server.http.headers.maxNumber	Sets the maximum number of HTTP headers permitted for any request submitted to the Device Manager web server.	Section 5.2.1.16

Table 5.1 Summary of Device Manager Property Files (continued)

Property	Description	Location
server.http.headers.maxLength	Sets the maximum length permitted for any HTTP header.	Section 5.2.1.17
server.http.entity.maxLength	Sets the maximum length of an HTTP request entity.	Section 5.2.1.18
server.http.log.reverseDNS	Flags whether the Device Manager server performs reverse-DNS (Domain Name Server) lookup for its access logging.	Section 5.2.1.19
server.http.cache.size	Sets the upper-limit size of the Device Manager server's internal file cache.	Section 5.2.1.20
server.http.cache.maxFileSize	Sets the maximum file size for server-side caching.	Section 5.2.1.21
server.http.fileTypes.noLog	Contains a comma-delimited list of the file types that are not logged in the Device Manager server's access log when being transferred via HTTP.	Section 5.2.1.22
Server.http.mode	This property sets whether the server is running in **real** mode or **simulation** mode.	Section 5.2.1.23
Server.installTime	This property contains the install date of Device Manager.	Section 5.2.1.24
Server.base.home	This property contains the installation directory of the common Component.	Section 5.2.1.25
dbm.driver	Designates the JDBC driver used for connecting to the Device Manager server's database.	Section 5.2.2.1
dbm.url	Specifies the connection string (database URL) for the Device Manager server's DBMS.	Section 5.2.2.2
dbm.logonID	Designates the logon ID for connecting to the Device Manager server's database manager.	Section 5.2.2.3
dbm.password	Designates the password for connecting to the Device Manager server's DBMS.	Section 5.2.2.4
dbm.traceSQL	Designates whether output SQL to trace.log or not	Section 5.2.2.5
dbm.simulation.url	Designates the URL of DBMS for the simulator.	Section 5.2.2.6
logger.loglevel	Determines the verbosity level of operational (trace) and error logging.	Section5.2.3.1
logger.MaxBackupIndex	Sets the number of rolling backups to keep of each log file before the oldest is erased.	Section 5.2.3.2
logger.hicommandbase.loglevel	Determines the verbosity level of operational (trace) and error logging which writes into HDvMtrace1.log by the common component	Section 5.2.3.3
logger.hicommandbase.sysloglevel	This property determines the verbosity level of operational (trace) and error logging which writes into the EventLog (for Windows) or the syslog (for Solaris) by the common component.	Section 5.2.3.4
logger.hicommandbase.MaxBackupIndex	This property sets the number of rolling backups to keep of HDvMtrace1 log file before the oldest is deleted.	Section 5.2.3.5

Table 5.1 Summary of Device Manager Property Files (continued)

logger.hicommandbase.MaxFileSize	This property sets the number of rolling backups to keep of HDvMtrace1 log file before the oldest is deleted.	Section 5.2.3.6
server.dispatcher.agent.priority	Assigns the priority for Device Manager service agent threads.	Section 5.2.4.1
server.dispatcher.message.timeout	Sets the timeout for pending response messages before they are expired (purged).	Section5.2.4.2
server.dispatcher.message.timeout.in.processing	Sets the timeout for processing messages (in minutes) those are not completed by some reason.	Section 5.2.4.3
server.dispatcher.daemon.pollingPeriod	Defines the polling interval for the background agents responsible for checking component status and configuration version.	Section 5.2.4.4
server.dispatcher.traps.purgePeriod	Defines the purging interval for stale SNMP traps or alerts.	Section 5.2.4.5
server.dispatcher.startTimeOfIgnoringConnectionAlert	Defines the start time of the interval for stopping SNMP communication alert.	Section 5.2.4.6
server.dispatcher.endTimeOfIgnoringConnectionAlert	Defines the end time of the interval for stopping SNMP communication alert.	Section 5.2.4.7
MIME Properties	Contains the translation/lookup table for all Multipurpose Internet Mail Extensions (MIME) file types recognized by the Device Manager web server.	Section 5.2.5
client.logger.trace	Defines whether to output the trace information or not	Section 5.2.6.1
client.message.timeout	Defines the maximum wait time for the Device Manager server response (timeout of connection) in seconds.	Section 5.2.6.2

5.2 Server Properties

The server properties are contained in the **server.properties** file, which is normally located in the **HiCommandServer/config** directory under the directory where the Device Manager server was installed.

The default directory on Windows® systems is:

C:\Program Files\Device Manager\HiCommand\HiCommandServer\config

On Solaris™ systems, the default directory for the configuration files is:

/opt/HiCommand/HiCommandServer/config

5.2.1 Server Web Configuration Properties

5.2.1.1 server.http.host

This property designates either the host name or the dotted-decimal IP address for this Device Manager web server. You should not need to change this setting under normal circumstances.

Default: localhost

5.2.1.2 server.http.socket.bindaddress

If the Device Manager server software is installed on a system with more than one network interface (multiple NICs), this property specifies the interface or interfaces where Device Manager should listen. If left blank (the default), Device Manager listens on all interfaces. Otherwise, enter the IP address of the specific interface on which Device Manager should listen.

Default: blank

5.2.1.3 server.http.port

This property assigns the port used for the Device Manager HTTP (web) server. The conventional port number used for a standard web server is **80**, but there may already be an Intranet server running on this port. Moreover, you should avoid low-numbered ports because these could conflict with other services installed on the server. As a general rule, you can pick any port between **1024** and **49151**.

Default: 2001

5.2.1.4 server.https.port

This property assigns the port used for the Device Manager secure HTTP web server. The conventional port number for a secure web server is **443**, but there may already be a secure Intranet server running on this port. As noted above, it is better practice to utilize a port number between **1024** and **49151** for a specialized (middleware) HTTP server. Be sure that it has a different value than the port designated for the HTTP listener.

Default: 2443

5.2.1.5 server.http.default

This property sets the name of the default index page for the Device Manager web server. If an HTTP request is made against a directory (for example, **https://hic.domain.com:2443/foo/**, where **foo** is a folder under the server's document root), the web server attempts to find and transfer a file named **index.html** to this directory. If none exists, a directory listing is returned to the client browser. Under normal conditions, you should not need to change the default value of this property.

Default: index.html

5.2.1.6 server.http.user

This property sets the user identity for the UNIX® user who owns the Device Manager server process. It is ignored when the server is running on a Windows® machine. You must start the Device Manager server as the root superuser on a UNIX® system, so that various listeners (for example, SNMP) can be started on ports below 1024. After starting these listeners, Device Manager changes the ownership of the process in which it is running to the user specified by this property. You should not need to change the default setting under normal circumstances.

Default: hicmd

5.2.1.7 server.http.group

This property sets the user group for the UNIX® user who owns the Device Manager server process (refer to the **server.http.user** property in section 5.2.1.6). You should not need to change this setting under normal circumstances.

Default: hicmd

5.2.1.8 server.http.chroot

This Boolean property flags whether to invoke a **chroot** (change root) command into the Device Manager server's installation directory. It is only applicable when you are running the operating system under a UNIX® operating system. Invoking the **chroot** command re-defines the location of the "/" directory to refer to the Device Manager server's installation directory, which limits the server's access to the file system. This limits the vulnerability of a system to outside attempts at infiltration, because rogue processes cannot damage anything outside the installation directory.

You may need to change this setting to **false** if the server's document root is specified as being located outside of the Device Manager installation directory. There should not be any need to change the default setting under normal circumstances.

Default: true

5.2.1.9 server.http.request.timeout

This property sets the read-blocking timeout of the HTTP socket connection (in milliseconds). It can be used to enable or disable the **SO_TIMEOUT** setting for client-connection sockets. Reading from the input stream associated with a socket will block for only this amount of time before the socket expires. Its default value is **5000** (5 seconds). A value of zero is interpreted as an infinite timeout, meaning that **SO_TIMEOUT** is disabled for client connections. This setting should only be changed by expert system administrators who want to fine-tune the server's performance.

Default: 5000 (5 seconds)

5.2.1.10 server.http.connection.priority

This property sets the priority for all client-connection threads spawned by HTTP requests made against the Device Manager server. Valid values are between **1** and **10** (**1** = minimum priority; **5** = normal priority; **10** = maximum priority). You should only modify this property if you are an expert System Administrator seeking to fine-tune the server's performance. Recommended values are between **5** and **8**.

Note: If the connection thread priority is set to **10** (maximum), any simultaneous request connections are queued for sequential processing, which defeats the purpose of a multi-threaded server. This setting would actually adversely affect server performance, particularly when you are loading complex HTML pages (for example, those containing many images).

Default: 7

5.2.1.11 server.http.connection.bufSize

This property sets the size (in bytes) for all of the server's input/output (I/O) buffers. Increased buffer size may improve request/response network performance for high-volume connections, while decreasing it can help reduce the backlog of incoming data. Do not set the default value smaller than 1024 bytes, or it can cause failure. You should only modify this property if you are an expert System Administrator seeking to fine-tune the server's performance.

Default: 8192 bytes

5.2.1.12 server.http.socket.backlog

This property assigns the maximum queue length for incoming connection indications (a request to connect), such as setting the **SO_MAX_CONN** attribute of the server socket. If a connection indication arrives when the queue is already full, the Device Manager server will refuse the new connection. You should only modify this property if you are an expert System Administrator seeking to fine-tune the server's performance.

Default: 50

5.2.1.13 server.http.socket.maxThreads

When a request has been issued and is being processed on the Device Manager server, a client has an active connection on the server. This property specifies the number of active *requests* that can be processed at one time on Device Manager server, not the maximum number of clients. Once this limit is reached, the next request will be dropped. You should only modify this property if you are an expert System Administrator seeking to fine-tune the server's performance.

Default: 50

5.2.1.14 server.http.socket.linger

This Boolean property toggles whether the **SO_LINGER** socket attribute is enabled for client connections with the Device Manager server. Setting this flag at its default value means that a *linger-on-close* timeout of 60 seconds is applied to socket connections. You should only modify this property if you are an expert System Administrator seeking to fine-tune the server's performance.

Default: true

5.2.1.15 server.http.socket.noDelay

This Boolean property toggles whether the **TCP_NODELAY** socket attribute is enabled for connections to the Device Manager server. Setting this flag at its default value disables the Nagle algorithm for TCP/IP packets. You should only modify this property if you are an expert System Administrator seeking to fine-tune the server's performance.

Default: true

5.2.1.16 server.http.headers.maxNumber

This property sets the maximum number of HTTP headers permitted for any request submitted to the Device Manager web server, and helps prevent certain types of denial of service and attempted buffer overflow attacks by restricting the effect of malicious requests containing a large number of headers. You should not need to change this setting under normal circumstances. The Device Manager server silently ignores any HTTP headers in excess of this number. Runtime errors are not automatically generated under such circumstances.

Default: 20

5.2.1.17 server.http.headers.maxLength

This property sets the maximum length permitted for any HTTP header (in bytes). You should not need to change this setting under normal circumstances. It helps prevent certain types of denial of service and attempted buffer-overflow attacks by restricting the effect of malicious requests that contain unusually large header fields. Headers longer than the specified length will be truncated by the Device Manager server without automatically generating runtime errors.

Default: 1024

5.2.1.18 server.http.entity.maxLength

This property sets the maximum length of an HTTP request entity (in bytes). You should not need to change this setting under normal circumstances. It helps prevent certain types of denial of service and attempted buffer overflow attacks by restricting the effect of malicious requests that contain unusually large payload entities. If the server detects a posted request longer than this value, it sends an error response to the client and logs details of the attempted request.

Default: 131072

5.2.1.19 server.http.log.reverseDNS

This Boolean property flags whether the Device Manager server performs reverse-DNS (Domain Name Server) lookup for its access logging. When set, this property causes the server to try to resolve the name of a client from the IP address for incoming connections. If the client's IP address can be resolved, the domain name is also written into the server's access log (for example, http://www.hds.com/193.36.36.6). Not all IP addresses on the Internet have an assigned domain name, so some logged requests might still be recorded as a numeric IP address, even with this flag turned on.

Note: While translation of the IP address to a domain name can assist analysis of the server's access logs, reverse-DNS lookups are expensive in terms of resources, and this feature may significantly degrade the server's performance, especially on a slow network. You should keep the setting at the default value for better performance.

Default: false

5.2.1.20 server.http.cache.size

This property sets the upper-limit size of the Device Manager server's internal file cache (in bytes). A value of **zero** turns file caching off, which may adversely affect server performance when delivering complex static files (HTML pages containing images, etc).

This setting could be increased on a host machine with sufficient RAM installed. However, since the number of static files being served by Device Manager is only in the order of a few pages, performance gains would most likely be quite trivial.

Default: 10000000 bytes

5.2.1.21 server.http.cache.maxFileSize

This property sets the maximum file size for server-side caching. Static files larger than this limit (for example, the GUI application JAR file) are read from disk instead of being cached. There is no significant difference in response time whether these files are cached in memory or read directly from disk. A **zero** value for this property turns file caching off, which may adversely affect the web server's performance.

Default: 1 MB

5.2.1.22 server.http.fileTypes.noLog

This property contains a comma-delimited list of the file types that are not logged in the Device Manager server's access log when being transferred via HTTP. Generally, logging should be performed only for the HTML pages being requested by a browser or other client, or the server's access log quickly becomes filled with entries for files such as graphics files, JavaScripts, or cascading style sheets.

The default value for this property eliminates logging for the majority of the resource-type files likely to be requested from the Device Manager web server. Whitespace in the list is ignored. If you want access logging for all files, set this property to **empty**.

Default: gif,jpg,jpeg,png,css,js

5.2.1.23 server.http.mode

This property sets whether the server is running in real mode or simulation mode. This property is only used for development of the application that is connected to Device Manager. You should not change this property for normal operation.

Default: real

5.2.1.24 server.installTime

This property contains the install date of the Device Manager.

Format : dd/mm/yyyy:HH:MM:SS ZZZZ (dd:day, mm:month, yyyy:year, HH:hour, MM:minute, SS:second ZZZZ (TimeZone)

Default: dd/mm/yyyy:HH:MM:SS ZZZZ (Install date)

5.2.1.25 server.base.home

This property contains the installation directory of the common component. This property is set by Device Manager installer. You should not change this property under normal circumstances.

Default: c:\Program Files\HiCommand\Device Manager\Base

5.2.2 Database Properties

The database properties configuration file contains the set of directives that pertain to establishing a connection with the Device Manager server's database. Before the Device Manager server will run you need to correctly enter these settings and start the Database Management System (DBMS). If the server cannot connect to its DBMS, an entry is written to the error log (the default location is in the **logs** directory). This information can help considerably when you are troubleshooting a new installation.

This file also holds debugging and optimization property settings for the Java™ Database Connectivity (JDBC) layer in Device Manager server. You should only modify these properties if you need detailed diagnostic information or if you are an expert System Administrator seeking to fine-tune certain aspects of performance and/or memory utilization.

5.2.2.1 dbm.driver

This property designates the JDBC driver used for connecting to the Device Manager server's database. Its default value will not need to be modified under the standard installation setup. The class file specified by this property is usually provided by the DBMS supplier, and must be on the Java™ classpath to Device Manager server's Java Runtime Environment™ (JRE™). This requirement is handled automatically during a normal installation of the server software.

Default: interbase.interclient.Driver

5.2.2.2 dbm.url

This property specifies the connection string (database URL) for the Device Manager server's DBMS, formatted as follows: **jdbc:subprotocol:subname**. The subprotocol string points to the location of the database file, which by default is located inside the **database/interbase** folder under the Device Manager server installation directory.

If this URL is incorrectly specified, the server will fail to start. If that occurs, examine the server's error log for an entry that looks something like the following:

interbase.interclient.UnavailableDatabaseFileException: [interclient][interbase] I/O error for file C:\HICOMMANDSERVER\DATABASE\INTERBASE\HICOMMAND.GDB Error while trying to open file. The system cannot find the path specified.

Default (Windows®):

//localhost/c:/Program Files/HiCommand/HiCommandServer/database/interbase/HiCommand.gdb

Default (Solaris™):

//localhost/opt/HiCommand/HiCommandServer/database/interbase/HiCommand.gdb

5.2.2.3 dbm.logonID

This property designates the logon ID for connecting to the Device Manager server's database manager. For security reasons, you might want to change the default ID after you create a new user in the Device Manager server's DBMS.

Default: sysdba

5.2.2.4 dbm.password

This property designates the password for connecting to the Device Manager server's DBMS. For security reasons, you may want to change this password after you either change the super-user password or create a new user for Device Manager in InterBase.

Default: masterkey

5.2.2.5 dbm.trace.SQL

This property designates whether the output is SQL or trace.log. Set **true** to output to SQL. Set **false** not to output to SQL.

Default: false

5.2.2.6 dbm.simulation.url

This property designates the URL of DBMS for the simulator. This property is not used in normal operation.

Default:
jdbc:interbase://localhost/C:/ProgramFiles/HiCommand/HiCommandServer/database/interbase/HiCommandSim.gdb

5.2.3 Logger Properties

This properties file contains a set of directives that configure Device Manager server's logging module, including the names, locations and verbosity level of operational and error logging of the various log files. You can also use this file to configure trace logging for debugging and diagnostic purposes.

5.2.3.1 logger.loglevel

This property determines the verbosity level of operational (trace) and error logging. The values accepted in this field are (in decreasing order of detail): **DEBUG**, **INFO**, **WARN**, **ERROR**, and **FATAL**. The default logging level for production systems is **WARN**, which means that warnings and error messages are written into the trace and error logs, but debugging or informational entries are not.

Default: WARN

5.2.3.2 logger.MaxBackupIndex

This property sets the number of rolling backups to keep of each log file before the oldest is deleted. If this property is set to zero, no rolling backups are created, and log files are simply truncated when their maximum file size is reached. When a log file reaches its maximum length its filename is modified by appending a counter, for example, **access.log.1**. As more backup log files are created, their counter or version suffix is incremented (for example, **access.log.1** becomes **access.log.2**), until the specified number of rolling backups have been created. After that, the oldest backup log file is deleted each time a new backup is created.

Default: 10

5.2.3.3 logger.hicommandbase.loglevel

This property determines the verbosity level of operational (trace) and error logging which writes into HDvMtrace1.log by the common component. Each logging event has its own importance level independent from its type (error, warning, information). The levels, in increasing order of importance, are: 30, 20, 10, and 0. The default logging level for production systems is 20, which means that messages for logging event levels 20, 10, and 0 are written into the HDvMtrace1.log, but messages for logging event level 30 are not.

Default: 20

5.2.3.4 logger.hicommandbase.sysloglevel

This property determines the verbosity level of operational (trace) and error logging which writes into the EventLog (Windows®) or the syslog (Solaris™) by the common component. Each logging event has its own importance level independent from its type (error, warning, information). The levels, in increasing order of importance, are: 30, 20, 10 and 0. The default logging level for production systems is 0, which means that messages for only the logging event leveled 0 are written into the EventLog (Windows®) or the syslog (Solaris™), but messages for the logging event leveled 30, 20, and 10 are not. The default value is recommended.

Default: 0

5.2.3.5 logger.hicommandbase.MaxBackupIndex

This property sets the maximum number of files to keep of the HDvMtrace1 log file before the oldest is deleted. Valid values are between 1 and 16. When a log file reaches its maximum length, its filename is modified by increasing a counter (e.g., for example, HDvMtrace2). As more backup log files are created, their counter or version suffix is incremented (e.g., HDvMtrace2.log becomes HDvMtrace3.log), until the specified number of rolling backups have been created. After that, the oldest backup log file is deleted each time a new backup is created.

Default: 10

5.2.3.6 logger.hicommandbase.MaxFileSize

This property sets the maximum size of each of the rolling backup Device Manager trace log files. The specified size is assumed to be in bytes unless you specify kB for kilobytes, MB for megabytes or GB for gigabytes. Valid values are between 4096 and 2147483647. Even if this directive is not found in the properties file, an internal default value of 1 MB will be used.

Default: 1 MB

5.2.4 Dispatcher Properties

This properties file contains a set of configurable directives pertaining to the operation of Device Manager server's dispatcher layer, including properties that allow the fine-tuning of various background processes (daemons) and the optimization of the thread-priority for service agents.

5.2.4.1 server.dispatcher.agent.priority

This property assigns the priority for Device Manager **Service Agent** threads. Valid values are between **1** and **10** (**1** = minimum priority; **5** = normal priority; **10** = maximum priority). You should change the default value only if you need to fine-tune the agent dispatcher's performance. The recommended values are between **5** and **8**, and you should *not* set it to the maximum thread priority (**9** - **10**), because while that may cause individual requests to execute faster, it is likely to cause an overall degradation in performance when multiple users are sending concurrent requests.

Default: 5

5.2.4.2 server.dispatcher.message.timeout

This property sets the timeout for pending response messages (in minutes) before they are expired (purged). A pending message consists of a response from a long-running process (for example, discovery of a storage array) that has not yet been either polled by the client or sent to the client via the Device Manager notification service.

Default: 15 minutes

5.2.4.3 server.dispatcher.message.timeout.in.processing

This property sets the timeout for processing messages (in minutes) that are not completed by some reason.

Default: 720 minutes

5.2.4.4 server.dispatcher.daemon.pollingPeriod

This property defines the polling interval (in minutes) for the background agents responsible for checking component status and configuration version. A value of zero will disable these polling agents.

Default: 5 minutes

5.2.4.5 server.dispatcher.traps.purgePeriod

This property defines the purging interval for stale SNMP traps or alerts (in minutes). A value of zero will disable the purging of traps from the server.

Default: 5 minutes

5.2.4.6 server.dispatcher.startTimeOfIgnoringConnectionAlert

This property defines the start time of the interval for stopping SNMP communication alert. Accessing the subsystem that is in regular reboot will occur this alert.

Default: 2:45

5.2.4.7 server.dispatcher.endTimeOfIgnoringConnectionAlert

This property defines the end time of the interval for stopping SNMP communication alert. If you access a subsystem that is in regular reboot, that will cause this alert.

Default: 3:15

5.2.5 MIME Properties

This file contains the translation/lookup table for all Multipurpose Internet Mail Extensions (MIME) file types recognized by the Device Manager web server. Each property in this lookup table maps a particular extension suffix to the MIME type for that file. You should not need to modify this setting under normal circumstances, and in any event only expert System Administrators should make any additions to this file.

5.2.6 Client properties

These files contain the configuration of Device Manager Web Client.

5.2.6.1 client.logger.trace

This property defines whether output the trace information or not by applying Java Web Start's log output function. Set **true** to output trace information. Set **false** to not output trace information.

Note: In order to output trace information, Java Web Start's log output function must be activated. See *HiCommand™ Device Manager Web Client User's Guide* MK-92HC001 for more information about Java Web Start's log output function.

Default: false

5.2.6.2 client.message.timeout

This property defines the maximum wait time for the Device Manager server response (timeout of connection) in seconds. The web client sends the notification messages to the server. The server sends the notification of the task complete or alert as the response this message. The connection between the client and the server is on while waiting the response of the notification from the server. This property sets the timeout of this waiting time. The client sends the notification message again after the timeout. This timeout will be applied each time Web Client accesses the server.

Note: When the client is accessing the server through a proxy server and the connection timeout of the proxy is shorter than the timeout of this property, the notification message may be lost, because the timeout of the proxy server cuts the connection before the Device Manager server can send the response to the Web Client. If this is the case, please set the timeout for this property to a time shorter than the timeout of the proxy.

Default: 300

Chapter 6 Troubleshooting

6.1 Problems and Solutions

Table 6.1 lists the common problems and solutions after installing the Device Manager on a Windows® or Solaris™ platform.

Table 6.1 General Troubleshooting Information (continues on the following pages)

Problem	Solution
DESCRIPTION: Cannot add 9900 subsystem. Failed during initialization (unable to find device). **Add Subsystem** command fails with error message: **Failed during initialization of SNMP connection at IP Address 192.215.46.115 Cannot connect to the device**. CAUSE: Improper IP address configuration.	Verify that you can ping the IP address from the 9900 subsystem's service processor (SVP). Verify SNMP connectivity and the community name string by executing the snmptest.exe file located in the `/Windows/Utilities/SNMP` folder of the Device Manager installation CD. Execute the utility from the command line in the format **SnmpTest host <community>**, where **host** represents the hostname or IP address of the remote host and **community** is the community string defined for the remote agent. The **community** parameter is optional, and has a default value of **public**. Verify the Hitachi SNMP Agent configuration, and make sure that the correct community name is specified in Device Manager (this should match community name that was entered when the Hitachi SNMP Agent was configured on the 9900 SVP). The Device Manager default community name is **public**. Make sure either that access by IP address is not limited by entries in the SNMP Setup Managers tab on the SVP, or if access is limited that the Device Manager server's IP address is added to the list of SNMP managers. For further information on Hitachi SNMP Agent configuration on the 9900 Remote Console system, please refer to the *Hitachi Lightning 9900 Remote Console User's Guide* (MK-90RD003). For more information on Device Manager requirements for the 9900 SVP and Remote Console, including Hitachi SNMP Agent, see *HiCommand™ Web Client Installation and Configuration Guide* (MK-91HC001).
DESCRIPTION: Inconsistencies in LUNs and Logical Group information. LUNs disappear or logical group information is inconsistent between Device Manager servers. CAUSE: Multiple Device Manager servers are managing the storage arrays.	Never have more than one active Device Manager server managing a single storage array at a time. Device Manager was designed to manage multiple storage arrays, but not to cooperate with other instances of Device Manager server to manage the same storage array. More than one active Device Manager client is not a problem.
DESCRIPTION: Not enough disk space for installation in a Windows® environment. CAUSE: The InstallAnywhere™ installer needs sufficient space to unpack the compressed installation files. InstallAnywhere™ uses the Windows® TEMP environment variable to locate the temporary directory for extracting the files. You should have at least 100 MB of free space available, preferably more, on this drive.	Choose a different drive and directory, or delete files from the Windows® TEMP directory to clear enough space for InstallAnywhere™.

Problem	Solution
DESCRIPTION: Cannot add 9900 subsystem. Failed during initialization (unable to find device). **Add Subsystem** command fails with error message: **Failed during initialization of SNMP connection at IP Address 192.215.46.115 Cannot connect to the device**. CAUSE: Improper IP address configuration.	Verify that you can ping the IP address from the 9900 subsystem's service processor (SVP). Verify SNMP connectivity and the community name string by executing the snmptest.exe file located in the `/Windows/Utilities/SNMP` folder of the Device Manager installation CD. Execute the utility from the command line in the format **SnmpTest host <community>**, where **host** represents the hostname or IP address of the remote host and **community** is the community string defined for the remote agent. The **community** parameter is optional, and has a default value of **public**. Verify the Hitachi SNMP Agent configuration, and make sure that the correct community name is specified in Device Manager (this should match community name that was entered when the Hitachi SNMP Agent was configured on the 9900 SVP). The Device Manager default community name is **public**. Make sure either that access by IP address is not limited by entries in the SNMP Setup Managers tab on the SVP, or if access is limited that the Device Manager server's IP address is added to the list of SNMP managers. For further information on Hitachi SNMP Agent configuration on the 9900 Remote Console system, please refer to the *Hitachi Lightning 9900 Remote Console User's Guide* (MK-90RD003). For more information on Device Manager requirements for the 9900 SVP and Remote Console, including Hitachi SNMP Agent, see *HiCommand™ Web Client Installation and Configuration Guide* (MK-91HC001).
DESCRIPTION: Device Not Supported. If the Config->Subsystems->New command fails with the error message: **The device is not supported by HiCommand™ Server. Description = Microsoft Corp. Windows 95** (a Windows 98® SVP may display a slightly different error message) CAUSE: You are trying to discover a 9900 subsystem and the IP Address is wrong, or the Hitachi SNMP Agent is not configured properly.	SOLUTION: Verify that the IP Address for the 9900 SVP is correct. If the IP Address is correct, the Hitachi SNMP Agent may not be correctly configured. With the SVP in modify mode, verify that the **Extension SNMP for Hitachi RAID Series** box on the **Install** tab of the SNMP Properties panel is checked, indicating that the Hitachi SNNP Agent program product is installed. For more information on Hitachi SNMP Agent installation and configuration on the 9900 subsystem see *Hitachi Lightning 9900 Remote Console User's Guide* (MK-90RD003). For more information on Device Manager requirements for the SVP and Remote Console, see *HiCommand™ Web Client Installation and Configuration Guide* (MK-91HC001).
DESCRIPTION: Solaris™ installation fails with the message: **InterBase Server was not properly installed. Try re-installing InterBase Server...** and a similar message displays about InterClient®. CAUSE: You may be trying to use the graphical installation mode without an adequate windowing environment on your system or without the executable **/usr/dt/bin/dtterm** in your path.	SOLUTION: Use the command line installation mode instead. (Refer to section 3.2).

Problem	Solution
DESCRIPTION: Cannot add 9900 subsystem. Failed during initialization (unable to find device). **Add Subsystem** command fails with error message: **Failed during initialization of SNMP connection at IP Address 192.215.46.115 Cannot connect to the device**. CAUSE: Improper IP address configuration.	Verify that you can ping the IP address from the 9900 subsystem's service processor (SVP). Verify SNMP connectivity and the community name string by executing the snmptest.exe file located in the `/Windows/Utilities/SNMP` folder of the Device Manager installation CD. Execute the utility from the command line in the format **SnmpTest host <community>**, where **host** represents the hostname or IP address of the remote host and **community** is the community string defined for the remote agent. The **community** parameter is optional, and has a default value of **public**. Verify the Hitachi SNMP Agent configuration, and make sure that the correct community name is specified in Device Manager (this should match community name that was entered when the Hitachi SNMP Agent was configured on the 9900 SVP). The Device Manager default community name is **public**. Make sure either that access by IP address is not limited by entries in the SNMP Setup Managers tab on the SVP, or if access is limited that the Device Manager server's IP address is added to the list of SNMP managers. For further information on Hitachi SNMP Agent configuration on the 9900 Remote Console system, please refer to the *Hitachi Lightning 9900 Remote Console User's Guide* (MK-90RD003). For more information on Device Manager requirements for the 9900 SVP and Remote Console, including Hitachi SNMP Agent, see *HiCommand™ Web Client Installation and Configuration Guide* (MK-91HC001).
DESCRIPTION: Solaris™ installation fails with the message: **Invocation of this Java Application has caused an Invocation Target Exception. This application will now exit.** CAUSE: You may be attempting to run the Device Manager **install.sh** program without a monitor attached to the system.	SOLUTION: Attach a monitor to the system, or telnet into the system and be sure to use the command-line installation. To run the installer as a text mode process, invoke the installer with the –i console parameter.

Problem	Solution
DESCRIPTION: Cannot add 9900 subsystem. Failed during initialization (unable to find device). **Add Subsystem** command fails with error message: **Failed during initialization of SNMP connection at IP Address 192.215.46.115 Cannot connect to the device**. CAUSE: Improper IP address configuration.	Verify that you can ping the IP address from the 9900 subsystem's service processor (SVP). Verify SNMP connectivity and the community name string by executing the snmptest.exe file located in the `/Windows/Utilities/SNMP` folder of the Device Manager installation CD. Execute the utility from the command line in the format **SnmpTest host <community>**, where **host** represents the hostname or IP address of the remote host and **community** is the community string defined for the remote agent. The **community** parameter is optional, and has a default value of **public**. Verify the Hitachi SNMP Agent configuration, and make sure that the correct community name is specified in Device Manager (this should match community name that was entered when the Hitachi SNMP Agent was configured on the 9900 SVP). The Device Manager default community name is **public**. Make sure either that access by IP address is not limited by entries in the SNMP Setup Managers tab on the SVP, or if access is limited that the Device Manager server's IP address is added to the list of SNMP managers. For further information on Hitachi SNMP Agent configuration on the 9900 Remote Console system, please refer to the *Hitachi Lightning 9900 Remote Console User's Guide* (MK-90RD003). For more information on Device Manager requirements for the 9900 SVP and Remote Console, including Hitachi SNMP Agent, see *HiCommand™ Web Client Installation and Configuration Guide* (MK-91HC001).

DESCRIPTION: Unable to run Disk Array Management Program 2 (DAMP) and the Device Manager server on the same system.
CAUSE: The Disk Array Management Program 2 (DAMP) from Hitachi, Ltd. for configuring HDS 9200 and Hitachi DF500 storage arrays require that version 1.2 of the Java Runtime Environment (JRE) be installed and be the default JRE.
The DAMP program fails to run if it detects that any other version of the JRE is configured as the default JRE. The Device Manager installation program installs version 1.3.1_03 of the JRE, thus causing the DAMP program to fail.
	SOLUTION:
	1. In Windows® Explorer or a drive window, navigate to the DAMP installation folder. (The default installation folder is "C:\Program Files\DA Manager GUI".)
	2. Right-click on the file `Startmgr2.bat` and choose **Edit**.
	3. Edit the file to make it functionally equivalent to the following:
	`@echo off`
	`set JAVA_HOME=″C:\Program Files\JavaSoft\JRE\1.2″`
	`set DAMP_ROOT_DIR_PATH=.`
	`set PATH=%JAVA_HOME%\bin;%PATH%`
	`java -classpath %JAVA_HOME% -jar CONFMNG2.JAR`
	4. Save the file and close your editor.
	5. Locate the icon on your desktop labeled Disk Array management program 2(GUI), right-click on it and select Properties. The shortcut properties dialog will display.
	6. On the **Shortcut** tab, in the Target edit box, go to the end of the displayed string by pressing the **End** key on your keyboard. Backspace over the ending double-quote character and the filename part of the referenced file "Confmng2.jar" (13 characters) and replace the filename with "Startmgr2.bat". Be sure to retype the double-quote character at the end of the string.
	7. In the **Run** selector, choose **Minimized**.
	8. Click **OK** to close the shortcut properties dialog.
	At this point you should be able to double-click the DAMP icon on your desktop to run the DAMP program.

DESCRIPTION: Incompatibility between Graph-Track 4.06 and Device Manager, such that installing one over the other can cause failures. CAUSE: Device Manager requires InterBase® (GDS32.dll) at core rev level 6.0.1.0 and Graph-Track 4.06 uses core rev level 5.6.0.29.	SOLUTION: Upgrade to Graph-Track 5.0 or higher If you are running Graph-Track 4.06, please see Graph-Track Alert #21 for installation instructions.

6.2 Using Trouble Information Acquisition (TIA) to Obtain Batch Error Information

When an error occurs in a Device Manager server, you can use TIA (Trouble Information Acquisition) to obtain information for analyzing errors. This information, including log files and a database, is obtained from the operating environment of the Device Manager server. You must log in with administrator or root access to use TIA.

6.2.1 Configuring TIA

To use TIA, you need to edit the TIA.properties file.

Open the command prompt or console window and navigate to the TIA installation directory. If necessary, change the values for the following directories coded in the TIA.properties file, as follows:

- **OUT_DIR** specifies the directory to which error information is output. This value must be specified when you use TIA.
- **SIM_DIR** specifies the directory where the RAID simulator is installed, if you use the simulator.
- **CLI_DIR** specifies the directory where CLI is installed, if CLI is installed on the machine where Device Manager server is installed.
- **HSCC_COMMONLOG_DIR** specifies the directory to which common component outputs the integrated trace log.
  Note: Do not change this value because it is written by Device Manager installer.
- **SYSLOG** specifies an absolute path for the event log file (in Windows®) or syslog (in Solaris™). The default value is as follows:
  - Windows®:

    c:/WINNT/system32/config/AppEvent.Evt
  - Solaris™:

    /var/adm/message

6.2.2 Using TIA to Acquire for Batch Error Information

When you use TIA to acquire batch error information, execute the following file:

Windows®:

<installation directory>/SupportTools/CollectTool/TIA.bat

Solaris™:

<installation directory>/SupportTools/CollectTool/TIA.sh

6.2.2.1 OUT_DIR Directory

After you execute TIA, the error information is output to the directory specified in **OUT_DIR** in the **TIA.properties** file. Figure 6.1 shows the subdirectory configuration, and Table 6.2 lists the files that are placed in the subdirectories.

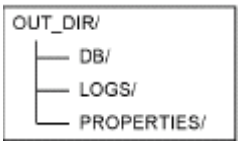

Figure 6.1 Subdirectory Configuration Under OUT_DIR

Table 6.2 Files Output to the OUT_DIR directory

File Name	Output Contents
keystore	The keystore file to be used when the security level of the Device Manager server is Secure Socket (SSL/TLS).
ActiveConnections.txt	Information about the active-connection on a server machine.
RunningServices.txt	Information about the services running on a server machine.
FileInfoList.txt	A list of all the files acquired by the TIA and update time of each file.

The character string (URL) of the DBMS connection-target database for the Device Manager server is written in a database file. All files in the directory containing the database file are output. Table 6.3 shows the files output to the OUT_DIR/DB subdirectory.

Table 6.3 Files Output to the OUT_DIR/DB Subdirectory

File Name	Output Contents
*.gdb	Database files.
*.gdb	Backup files of database files.

Table 6.4 Files Output to the OUT_DIR/LOGS Subdirectory

Log File Type	File Name	Output Contents
Device Manager server	version	Version information about the operating environment for Device Manager server.
	Boot.log	Boot information about Device Manager server in Solaris™.
	access.log	HTTP access log information for Device Manager server.
	service.log	XML API log information for Device Manager server. An XML API name and its status is output.
	trace.log	Trace log information for Device Manager server.
	error.log	Error information for Device Manager server. The error information extracted from the trace log is output.
	HDvMtrace*.log	Trace log information for Device Manager server that is output by the common component. The asterisk (*) in the file name indicates a file number.
	hntr2*.log	Integrated trace log information that is output by the common component. The asterisk (*) indicates a file number.
	apilog	API trace information executed by DAMP library.
	svplog	Communication trace information between the DAMP library and the SVP.
	dpTrcLog.log	Communication trace information between a Device Manager server and DAMP-API.
	cim_access.log	Access log information from Device Manager server to the CIM-WBEM service interface.
	The file name specified in the SYSLOG field in the TIA.properties file	System log information for the operating system.
Installation	HiCommand_Device_ Manager_*.*_ InstallLog.log	Log information during the installation of Device Manager server. The asterisks (*.*) of the file indicates the version and revision of the Device Manager server.
	HDvM_OldInstallLog_*.*_ n.log	When an overwrite installation is performed, the old installation log is saved as this file name. *.*_n indicates a version, revision, and build number of the Device Manager server respectively. (for example, HDvM_OldInstallLog_2.2_08.log)
	HICOMMAND_DeviceMan ager_UninstallError.log	Error log information output during the uninstallation of Device Manager server.
CLI **(Note 1)**	HiCommandCLI.log	CLI trace log information for Device Manager server.
	MessageTrace.log	XML API request or response information for Device Manager server CLI.
RAID Simulator **(Note 2)**	simulator.log	Log files for RAID Simulator.
	RAIDSimConfig.log	Log information for a configuration tool of RAID Simulator.

Note 1: The log information is obtained only when CLI_DIR is configured in the TIA.properties file.

Note 2: The log information is obtained only when SIM_DIR is configured in the TIA.properties file.

The properties files are output to the Device Manager properties directory. For more information on Device Manager properties, see Chapter 5.

Table 6.5 Files Output To The OUT_DIR/Properties Subdirectory

Properties File	Output Contents
server.properties	Web server function properties
database.properties	Database-related properties
logger.properties	Log-related properties
dispatcher.properties	Thread-related properties
mime.properties	MIME-related properties
security.properties	Security-related properties
wbemservices.properties	Web client related properties
HiCommandCLI.properties ***Note1***	Device Manager server CLI properties
simulator.properties ***Note 2***	RAID Simulator properties.
RAIDSimConfig.properties ***Note 2***	Configuration tool properties for RAID Simulator
import.properties ***Note 2***	Properties referenced when a data file is created for RAID Simulator

Note 1: This information is obtained only when CLI_DIR is configured in the TIA.properties file.

Note 2: This information is obtained only when SIM_DIR is configured in the TIA.properties file.

6.3 Using hcmdsras to Obtain Common Web Service and Single Sign-On Service Error Information

If either the common web service or single sign-on service does not start, you can use the **hcmdsras** command to acquire error information.

6.3.1 Acquiring Windows® Error Information

Command:

<installation directory>\Base\bin\hcmdsras.bat

Maintenance information:

<installation directory>Base\hcmdsrasdata

Table 6.6 Files Acquired By the hcmdsras Command (Windows®) (continues on the follow pages)

Folder/File Name	Description
installation-folder\hcmdsrasdata	
\c4web	This folder holds a SOAP engine for the common component.
\conf	This folder holds a configuration definition file.
c4websv.cfg	This is a configuration definition file
\logs	This folder holds log files.
SOAP-service-name-Number.log	These are log files
SOAP-service-name--aplog-0.log.lck	
SOAP-service-name-aplog-0.log	
\conf	This folder holds configuration definition files for the common component.
hcmdsrepClient.dtd	These are configuration definition files
hcmdsrepServer.dtd	
hcmdswebpp.ini	
hsso.conf	
HssoServerRes1_0.dtd	
init.conf	
javavm.properties	
setup.iss	
usrconf.properties	

Table 6.6 Files Acquired By the hcmdsras Command (Windows®) (continued)

Folder/File Name	Description
\database	This folder holds a backup file for the common component database.
hbase_backup.gbk	This is a backup file for the common component database
\httpsd	This folder holds the common web service.
\conf	This folder holds configuration definition files.
httpsd.conf	These are configuration definition files
mime.types	
\logs	This folder holds log files.
access.xxx	These are log files for the common web service
error.xxx	
httpd.pid	
hws.trcid	
\sslc	This folder holds an SSL tool.
\bin	
\demoCA	
sslc.cnf	This is an SSL configuration definition file
\hwc	This folder holds web container servers for the common component.
\containers	This folder is a work folder for each web container server.
\server-name	This is a directory for each web container server
\logs	This folder holds log files.
hwc_containernumber.log	These are log files for the container server
hwc_exceptionnumber.log	
hwc_maintenancenumber.log	
hwc_messagenumber.log	
hwc_shutdownnumber.log	
servletnumber.log	
\usrconf	This folder holds user definition files.
javavm.properties	These are user definition files
usrconf.properties	
web-users.xml	
javacorexxx.xxxx.txt	This is a JavaVM thread dump
hs_err_pidxxxx.log	This is a JavaVM log (xxxx: process ID)

Table 6.6 Files Acquired By the hcmdsras Command (Windows®) (continued)

Folder/File Name	Description
\redirector	This folder holds a redirector.
\logs	This folder holds a log file.
hws_redirectnumber.log	This is a log file for a redirector
mod_jk.conf	This is a redirector definition file
workers.properties	This is a worker definition file
\log	This folder holds log files for the common component.
hcmdsRepositorynumber.log	These are log files for the common component
hcmdsRepositoryVer	
hcmdssrvnumber.log	
hcmdswebnumber.log	
hcmdsPPname.log	
HssoServernumber.log	
HssoServerVer	
RepClientnumber.log	
RepServletnumber.log	
RepServletVer	
RepClientVer	
hcmdssupnumber.log	
hcmdssup_ver	
service-name_err.log	This is a standard-output log file for JavaVM
service-name_out.log	This is a standard error log file for JavaVM
each-service-nameNumber.log	This is a log file for service control
\sample	This folder holds files that you must not overwrite.
\conf	This folder holds configuration definition files.
build	This file contains version information for the common component.
hcmdsrepClient.dtd	These are configuration definition files for the common component
hcmdsrepServer.dtd	
hsso.conf	
HssoServerRes1_0.dtd	
init.conf	
javavm.properties	
setup.iss	
usrconf.properties	

Table 6.6 Files Acquired By the hcmdsras Command (Windows®) (continued)

Folder/File Name	Description
\spool	This folder holds an integrated trace log for the common component.
hntr2number.log	This file contains an integrated trace log for the common component
dirlist	This file contains a list of directories under HiCommand-Suite-Commmon-Component-installation-directory.
hcmdsgetname	This file contains the names of programs registered for usage information.
hcmdsist.log	This file contains an installation log for the common component.
hcmdsrtn.inst	This file contains an installation log for the common component.
hcmdsuit.log	This file contains an uninstallation log for the common component.
hcmdsrtn.uit	This file contains an uninstallation log for the common component.
ipconfiglist	This file contains the results acquired by the ipconfig command.
netstatlist	This file contains the results acquired by the netstat command.
servicelist	This file contains a list of services.

6.3.2 Using hcmdsras to Acquire Solaris™ Error Information

Command:

/opt/HiCommand/Base/bin/hcmdsras

Maintenance information:

/var/opt/HiCommand/Base/hcmdsrasdata.tar

Table 6.7 Files Acquired By the hcmdsras Command (Solaris™) (continues on the following pages)

Directory/File Name	Description
/var/opt/HiCommand/Base/hcmdsrasdata	
/c4web	This folder holds a SOAP engine for the common component.
/conf	This folder holds a configuration definition file.
c4websv.cfg	This is a configuration definition file
/logs	This folder holds log files.
SOAP-service-name-Number.log	These are log files
SOAP-service-name-aplog-0.log.lck	
SOAP-service-name-aplog-0.log	
/logs_opt	This folder holds log files.
SOAP-service-name-Number.log	These are log files
SOAP-service-name-aplog-0.log.lck	
SOAP-service-name-aplog-0.log	
/conf	This folder holds configuration definition files for the common component.
hcmdsrepClient.dtd	These are configuration definition files
hcmdsrepServer.dtd	
hcmdswebpp.ini	
hsso.conf	
HssoServerRes1_0.dtd	
init.conf	
installed.cnf	
javavm.properties	
usrconf.properties	
/database	This folder holds a backup file of a database for the common component.
HBASE_BACKUP.gbk	This is a backup file of a database for the common component

Table 6.7 Files Acquired By the hcmdsras Command (Solaris™) (continued)

Directory/File Name	Description
/httpsd	This folder holds the common web service.
conf	This folder holds configuration definition files.
httpsd.conf	These are configuration definition files
mime.types	
/logs	This folder holds log files.
access.*xxx*	These are log files for the common web service
error.*xxx*	
httpd.pid	
hws.trcid	
/sslc	This folder holds an SSL tool.
/bin	
/demoCA	
sslc.cnf	This is an SSL configuration definition file
/hwc	This folder holds web container servers for the common component.
/containers	This folder is a work folder for each web container server.
/*server-name*	This is a directory for each web container server
/logs	This folder holds log files.
hwc_container*number*.log	These are log files for the container server
hwc_exception*number*.log	
hwc_maintenance*number*.log	
hwc_message*number*.log	
hwc_shutdown*number*.log	
servlet*number*.log	
/usrconf	This folder holds user definition files.
usrconf.properties	These user definition files
usrconf.sh	
web-users.xml	
server-name.log	A standard log file output when the container server starts
server-name.sh	This file contains a shell script to start a web container server.
*server-name*_stop.sh	This file contains a shell script to stop a web container server.
javacore*xxx.xxxx*.txt	This is a JavaVM thread dump
hs_err_pid*xxxx*.log	This is a JavaVM log (*xxxx*: process ID)

Table 6.7 Files Acquired By the hcmdsras Command (Solaris™) (continued)

Directory/File Name	Description
/redirector	This folder holds a redirector.
/logs	This folder holds log files.
redirector_initialize.log	This is a log file for a redirector
mod_jk.conf	This is an environment configuration file for a redirector
workers.properties	This is a worker definition file
/HNTRLib2	This is an integrated trace log for the common component
/spool	This is a directory
hntr2*number*.log	This is an integrated trace log file
/interclient	This folder holds InterClient information.
VERSION.TXT	This is an InterClient version information file
/interbase	This folder holds InterBase information.
VERSION.TXT	This is an InterBase version information file
/log	This folder holds log files for the common component.
hcmdsRepository*number*.log	These are log files for the common component
hcmdsRepositoryVer	
hcmdssrv*number*.log	
hcmdsweb*number*.log	
hcmdsPPname.log	
HssoServer*number*.log	
HssoServerVer	
RepClient*number*.log	
RepServlet*number*.log	
RepServletVer	
RepClientVer	
hcmdssup*number*.log	
hcmdssup_ver	

Table 6.7 Files Acquired By the hcmdsras Command (Solaris™) (continued)

Directory/File Name		Description
	varlist	This file contains a list of directories under /var/opt/HiCommand.
	optlist	This file contains a list of directories under /opt/HiCommand.
	pslist1	These files contain a list of services.
	pslist2	
	vmstatlist	This file contains the result acquired by the vmstat command.
	netstatlist	This file contains the result acquired by the netstat command.
	hcmdsgetname	This file contains the result acquired by the hcmdsgetname command.
	build	This file contains version information for the common component.
	hcmdsinst.log	This file contains an installation log for the common component.
	hcmdsrtn.inst	This file contains an installation end code.
	hcmdsuit.log	This file contains an installation log for the common component.
	hcmdsrtn.uit	This file contains an uninstallation log for the common component.

6.3.3 Acquiring a Thread Dump

If Device Manager uses the single sign-on function and one of the following events occurs, collect a JavaVM thread dump to check the cause of the problem:

- The Device Manager login window is not displayed when you start Web Client.
- The Device Manager main window is not displayed after logging on to Device Manager.
- The Device Manager main window is not displayed when you start a Device Manager server from Tuning Manager.

To acquire a JavaVM thread dump:

Windows®:

1. In **<installation directory>\Base\hwc\containers\HiCommand**, create a file called **dump**.
2. From the service window, stop single sign-on service.
3. The **javacorexxx.xxxx.txt** file is output to **<installation directory>\Base\hwc\containers\HiCommand**.
4. From the service window, start single sign-on service.

Solaris™:

1. Execute **kill -3 PID**. PID is a process ID of single sign-on service written in the **/var/opt/HiCommand/Base/tmp/HiCommand.pid** file.
2. The **javacorexxx.xxxx.txt file** is output to **/opt/HiCommand/Base/hwc/containers/HiCommand.**
3. Execute the following command to stop the single sign-on service process: **/etc/init.d/hicommand-SSOS stop**
4. Execute the following command to start a single sign-on service process: **/etc/init.d/hicommand-SSOS start**

6.4 Contacting the Hitachi Data Systems Support Center

If you need to call the Hitachi Data Systems Support Center, be sure to provide as much information about the problem as possible, including the circumstances surrounding the error or failure, and the exact content of any error messages.

The worldwide Hitachi Data Systems Support Centers are:

- Hitachi Data Systems North America/Latin America
  San Diego, California, USA
  1-800-348-4357
- Hitachi Data Systems Europe
  Contact Hitachi Data Systems Local Support
- Hitachi Data Systems Asia Pacific
  North Ryde, Australia
  011-61-2-9325-3300

Glossary, Acronyms, and Abbreviations

DASD	direct access storage device
DKCMAIN	The version of microcode running on a 9900 subsystem.
GB	gigabyte(s)
kB	kilobyte(s)
LAN	local-area network
LDEV	logical device
LUN Manager	A 9900 and 9900V subsystem software option that allows users to add and delete SCSI paths, set host and port modes, configure the fibre-channel topology, and set or release high-speed mode. See the *Hitachi 9900 LUN Manager User's Guide* (MK-91RD049) or the *Hitachi 9900 V Series LUN Manager User's Guide* (MK-92RD105) for more information.
SANtinel™	A 9900 subsystem software option that allows the user to set or change security parameters for one or more ports, set or change LUN group definitions, and set or change WWN or WWN group parameters. See the *LUN Manager User's Guide* (MK-91RD049) for more information.
SNMP	Simple network management protocol (part of the TCP/IP protocol suite). ***Note:*** The Hitachi SNMP Agent is a program product that must also be installed on the SVP and on each 9900 subsystem. For more information on Hitachi SNMP Agent on the 9900, see the *Remote Console User's Guide* (MK-90RD003).
SVP	The service processor on a 9900 or 9900V series subsystem. This is the laptop mounted on the front of the unit (inside the cabinet).
TCP/IP	transmission control protocol/internet protocol
WWN	Worldwide name, which is a unique identifier for a particular open-system host bus adapter port consisting of a 64-bit physical address (the IEEE 48-bit format with 12-bit extension and 4-bit prefix).

Index