HITACHI
Inspire the Next

Using Hitachi HiCommand® Device Manager Software and Best Practices Guide

Application Brief

By Mike Le Voi, Bob Curtin, and John Harker

March 2007

Hitachi Data Systems

Executive Summary

This application brief discusses a number of aspects of Hitachi HiCommand® Device Manager software that will make working with the application a little easier. It also includes some Best Practices collected from the field. As of this writing, the GA version of Device Manager software is 5.6, and all statements within this paper apply to that level, unless otherwise noted.

Contents

Using Hitachi HiCommand® Device Manager Software and Best Practices Guide

Application Brief

By Mike Le Voi, Bob Curtin, and John Harker

Getting Started

Terminology

This section provides an explanation of terms describing a Hitachi HiCommand® Device Manager software configuration. There are a minimum of two parts to the design of a Device Manager software installation—the Device Manager Server and the Device Manager Client. There is a third optional factor, which is the Device Manager Host Agent. You only have to install the Host Agent if you wish to control your replication environment with Device Manager software or you want to provision storage with Device Manager software's Provisioning Manager component. In all other cases, it is not a requirement that a Host Agent be installed, although it is recommended.

The Device Manager Server is simply the server where the main application is installed. It will contain all the code that is necessary to use the product. The Client is a browser you use to connect to the Device Manager Server. The Client can be any machine that has network connectivity to the Device Manager Server. It can be the Device Manager Server. There can be any number of "clients" connected to the server at any given point in time. The only requirement of the client is that you are running a supported browser for most functions of the Device Manager graphical user interface (GUI).

Please make sure you check the supported browsers for your version (for example, Firefox is not supported with a Microsoft® Windows client, but it is supported with Sun Solaris and HP-UX). For certain Device Manager GUI functions, such as the physical view of certain storage systems, or Resource Group Management, you need to install the appropriate Java Runtime Environment (JRE) on the client machine. Again, please verify the correct JRE is installed for the version of Device Manager software you are using (Hitachi Data Systems does not currently support JRE 1.5 for use with the Device Manager Client).

Device Manager Host Agents, while not required, do give the user more functionality. The initial benefit they provide is to make a connection between a WWN and a host name. When LUNs are secured to a host it is done with the WWN, and that is the information that is kept by the storage system. The storage system does not keep track, nor does it care about, the name given to the server. Most customers do care about this name and when they look at their configuration they expect to see the correct name in the host view. The only way the name can be included in the host view is if it is entered manually (either through the GUI or more preferably with the command line interface) or by the running of Device Manager Host Agents. There is more information about agents in a later section.

There is one other point relative to terminology. HiCommand is not Device Manager software. Hitachi Data Systems offers the HiCommand Suite of products; HiCommand Device Manager is just one of the many products in the suite.

Starting Steps

Initial procedures for using Device Manager software can be summarized by the following steps:

1. Install the software and license, and then log in.
2. View host information if collected by agents; add hosts manually, if necessary.
3. Create logical group hierarchy for storage groups.
4. Add connected storage systems to be managed.
5. LUN SCAN the storage systems. Move storage to correct logical groups.
6. Create administrative users and assign permissions.
7. Create and allocate resource groups, if required.
8. Perform storage operations.

Installation

An installation of Device Manager software will need to meet a number of specific requirements in order to run properly. There are minimum needs for space, memory, platform, etc. In most cases, something higher than the minimum is always a good move (memory and more memory). While it may not be an initial concern, lack of sufficient memory can cause performance problems.

The number of resources that can be effectively managed, as defined by the Server Installation Manual, should be viewed as an absolute high water mark. If your customer is anywhere near those numbers you will want to consider two (or more) Device Manager Servers. Also, this product scales a lot better with Windows, currently, than with Solaris.

A good idea is to create a backup and an exported database right after the new install is completed. Since you have not done anything with the product, you are in effect creating a copy of nothing, but it can prove useful if you wish to start over. If you have made some changes to the configuration and you do not like those changes, you can restore your “empty” database and it will be as if you just installed the product from scratch. Without this backup, you will need to install the product from scratch to get back to the beginning.

So what are “Best Practices” for a Device Manager install or upgrade? A new install needs only adhere to the stated requirements and it should run fine. Those requirements are complicated in a Solaris and Linux environment by the need to modify kernel settings. Just remember that these settings are all additive. So, if there are existing values you need to add the HBase and Device Manager numbers to them.

If HiCommand Tuning Manager software is also being installed there is more addition that needs to be done. Always install Tuning Manager software before Device Manager software if they are on the same machine. In general, however, it is recommended that Tuning Manager and Device Manager be installed on separate physical servers or Windows “Virtual Machines.”

If you are looking at an upgrade, you should always back up and export the databases before starting. While this backup is hardly ever needed, when it is you will be glad it is there. The biggest upgrade is when you are going from Device Manager 3.5 or earlier to 4.0 or later. This is due to the change in the underlying database from InterBase to HiRDB (Hitachi Relational Database).

During the upgrade you are asked if you want to migrate the database during this install or later on. We have seen a lot more problems when the migration is done after the upgrade. Please remember that this type of upgrade needs to be done on the machine that is running the older version of Device Manager software. This is because you need InterBase and InterClient up and running on the server in order to do the migration.

If you are running Device Manager 4.x or higher, and you are upgrading to a new version on a new server, you must do an export of the database (if you wish to retain that information). A backup of the Device Manager Server can only be restored on the same machine, in the same location. If you are moving, you will need to import the database to the new platform. That import will also work to any platform. You can export from Windows and import to Solaris and vice versa.

One final note on upgrading: Upgrades are allowed to any version that is equal to or greater than the one that is currently installed. You cannot downgrade. To go to an earlier version of the product you need to remove the installed edition.

Host Agents

The Device Manager Host Agent is provided to supply current volume information about a host. The agent is not required to run Device Manager software unless you wish to collect the data that it can gather, run Device Manager's Provisioning Manager component, or control your replication configuration with Device Manager.

Installation of the Host Agent is fairly simple and only requires you to enter the IP/Hostname of the Device Manager Server, a port (by default 2001), a userid, a password, and an execution schedule (set up by a separate program called Device ManagerAgt_schedule).

While the amount of overhead produced by this product is minimal, it is recommended to run the collection routine only once a day (and they can be run manually at any time). Additionally, if you have numerous Host Agents installed, you will want to stagger their reporting times so they are not all trying to connect to the Device Manager Server at the same time.

So what makes up the Host Agent? Basically, there is the Hitachi Logical Device Utility (Hldutil), Hitachi Storage Assessment Tool, and the agent daemon. Hldutil uses SCSI commands to get information on each of the connected LUNs. Below is some of the information that is collected:

Port WWNs

- Node WWNs
- LDEV
- File system name
- Port
- Server name
- LU usage
- Vendor name
- Serial number
- LUN
- Target ID
- Mount point
- SCSI address
- LU capacity
- Pair status
- Subsystem (storage system) type

The most interesting metrics are the server name, the LU capacity, and pair status. The first is useful since this data can be used to connect the customer's server name to the WWN. Without running the Host Agent, the only other way to make this connection is for someone to manually type in this relationship. As for the capacity, many customers will use this data to bill their users. Finally, the pair status will help the Device Manager Server manage a replication environment (which may include a version of Hitachi TrueCopy® Remote Replication software, a version of Hitachi ShadowImage™ In-System Replication software and Hitachi Copy-on-Write Snapshot software).

The second facility used by the Host Agent is called the Hitachi Storage Assessment Tool. This feature launches the logical device utility, connects to the Device Manager Server, transmits an XML file built from the results of the utility, and confirms its receipt at the Server. This is the process that is set up when you install the application and run the schedule program.

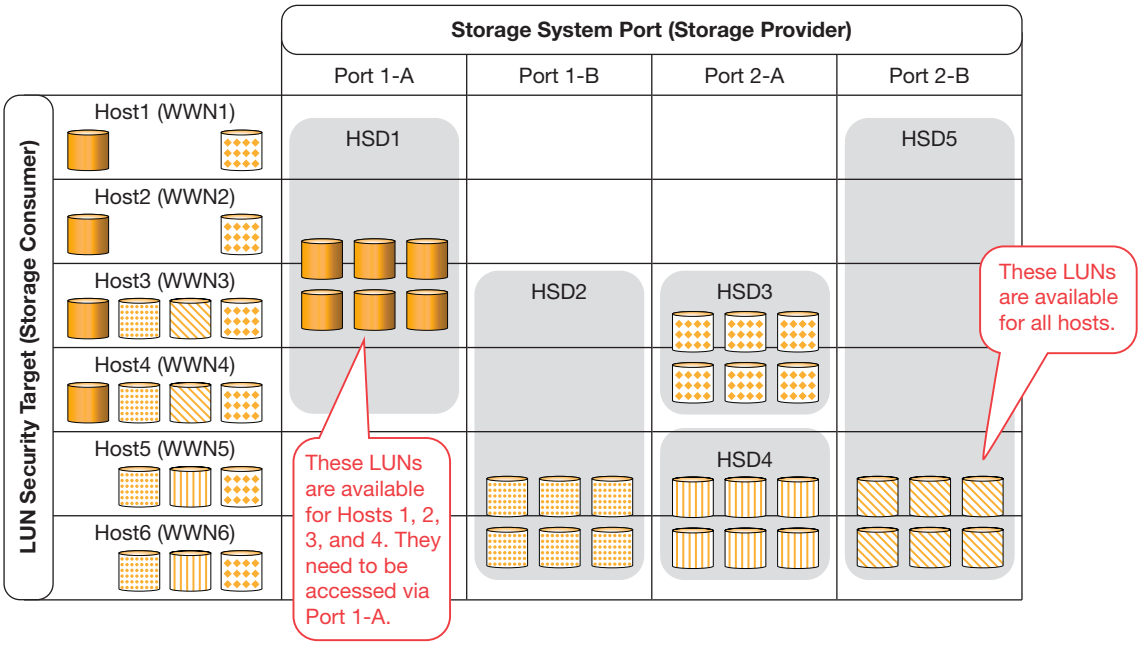
Host Storage Domains

What are Host Storage Domains?

Host storage domains (HSDs) are a feature of Hitachi storage control units. This feature enables the attachment of 1024 heterogeneous Fibre Channel host ports to one physical storage port on the Hitachi Universal Storage Platform or Network Storage Controller control unit. (This feature is also available on Hitachi Adaptable Modular Storage and Workgroup Modular Storage systems with 128 virtual ports per physical port.) Each Fibre Channel host connection comes through a virtual port and is assigned its own address space or host storage domain, which cannot be seen or accessed by any other virtual port. This provides the scalability of host connections and safe multitenancy, which is required when many applications share the same physical resources in a virtualized environment. An HSD can be shared with another virtual port on a different physical port for alternate path support. Additionally:

- An HSD can be defined on each Fibre Channel port on storage systems.
- From the LUN security management perspective, an HSD is a (logical) object to manage LUN and WWN combination under (physical) Fibre Channel port.
- HSD groups include WWNs, which are accessible through the Fibre Channel port to which the HSD belongs.
- HSD also defines SCSI target ID, which is recognized by the Host. So, by configuring appropriate target ID and assigning appropriate WWNs and LUNs on the HSD, the storage administrator can realize the LUN security configuration that he or she wants.
- Since a WWN cannot be used multiply under a Fibre Channel Port, a LUN security Matrix, as shown in Figure 1, must to be considered.

Figure 1. LUN Security Matrix—Acceptable HSD Usage

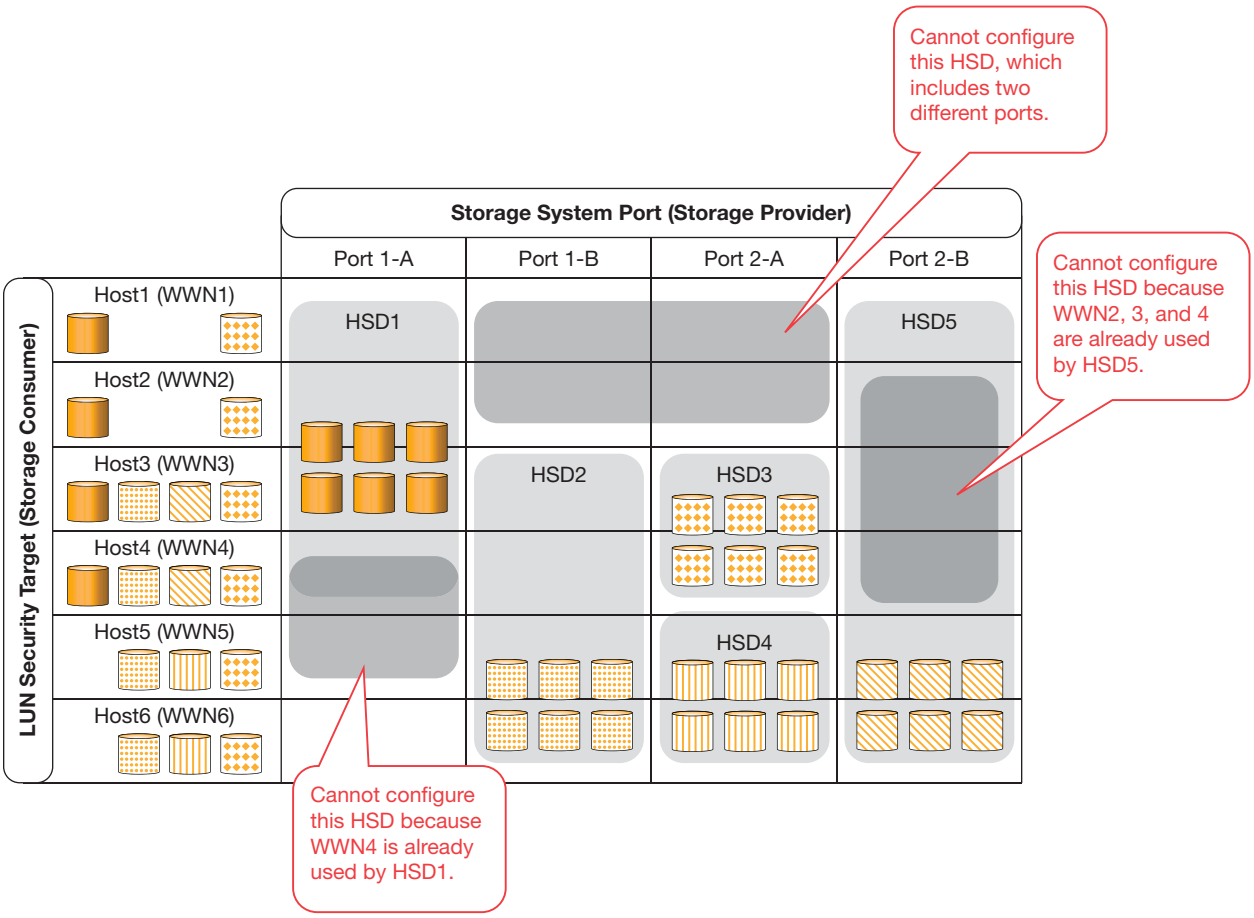


This example of a LUN Security Matrix for a Hitachi Storage System depicts the acceptable use of Host Storage Domains and the ports that may be used to access the LUNs.

Further, note:

- The same WWN cannot be configured on different HSDs under same Port, as shown in Figure 2.
- One HSD cannot be defined as involving two different ports.

Figure 2: LUN Security Matrix—Unacceptable HSD Usage



A new HSD may not be configured if a WWN is already in use by another HSD, and an HSD may not include two different ports.

Here is an example of the use of HSDs. Suppose you have two hosts, Host A and Host B. You choose to assign LDEV 0:00 for both hosts (as a shared volume) as well as to assign LDEV0:01 only for HOST A and LDEV 0:02 only for HOST B. Assume each Host has two HBAs and HiCommand Dynamic Link Manager software is being used. What shall you do?

The Best Practice is to configure four individual HSDs. Because there are three different WWN combinations to realize the LUN security that you want to configure, you have to map the LDEVs as follows:

Storage System Port 1

1. HBA 1 for Host A (HSD1) – LDEV 0:00 and 0:01
2. HBA 1 for Host B (HSD2) – LDEV 0:00 and 0:02

Storage System Port 2

1 HBA 2 for Host A (HSD3) – LDEV 0:00 and 0:01

2 HBA 2 for Host B (HSD4) – LDEV 0:00 and 0:02

The Best Practice is to "name" HSD1 and HSD3 with the same name. Most people name the HSD with the same name as the Host. The same applies to HSD2 and HSD4.

The HiCommand Device Manager GUI provides an "Add Storage Wizard" for storage administrators to ease complexity of HSD usages. The Wizard detects the need to create a new HSD when the Storage Administrator tries to configure a new LUN security. Alternately, if it does not have to, Device Manager software reuses already defined HSD for the new LUN configuration. Device Manager software also lets Storage Administrators know which ports on storage system are still available for LUN security configuration, which is trying to be configured.

Logical Groups, Storage Groups, Resource Groups

A default logical group called All Storage is built during initial configuration by the Add Subsystem function. You (or your administrator) can create your own logical structure, reflecting names and subdivisions of storage meaningful to administration tasks. Creation of different administrative users allows for logging and individual authorization profiles at different levels.

You can create your own logical hierarchy for storage groups, or use the group hierarchy created by the LUN Scan operations and reconfigure it as needed. Storage groups can be nestled within logical groups, or at the top level as needed. A logical group cannot be nested within a storage group.

A storage group is a user-defined set of LUNs that can be manipulated as a group. When you view the contents of a storage group, you can view property information on the selected storage group and information on the Logical Devices (LDEVs) managed under that storage group. You can add, remove, and move storage in a storage group, and you can change the security properties of a storage group.

There are two Device Manager requirements for Logical Groups:

- If a group contains subgroups, that group cannot contain any LUNs.
- Any one group can contain only LUNs for a single storage system.

How do you use logical groups and storage groups? The Best Practice is to make the Logical Group structure simple and consistent. The structure should fit the natural method that you have for provisioning and administration. Here are two simple examples:

Example 1. Site-based Structure

Site A

Windows Hosts

- Host WIN1
  - Storage for storage system A
  - Storage for storage system B
  - Etc.

HP-UX Hosts

- Host HP1
  - Etc.

Site B

- Etc.

Example 2. Responsibility-based Structure

Finance

- Host Fin1
  - Storage for storage system A
  - Storage for storage system B
  - Etc.
- Host Fin2

Payroll

- Host Pay1
  - Etc.

Choose something that is convenient for you. You can use CLI features to automate the maintenance of this structure if it resembles either of the schemas shown above.

When upgrading from Device Manager 4.3 or earlier, the user roles and properties set for administrative users are converted into the permissions, groups, and properties supported in version 5.0 or later.

Physical View Operations

Clicking on the Physical View button will mean different things for various storage systems. In the case of the Hitachi Thunder 9200™ and Thunder 9500™ V Series modular storage systems, the Hitachi Lightning 9900™ Series and Lightning 9900 V Series enterprise storage systems, and the Sun StorEdge T3 systems, when you click on that button you are submitting a request to the Device Manager Server to display a window showing the physical configuration of that storage system with data garnered from the Device Manager database. This process is handled by a Java application, and because of that it needs a supported JRE installed on the client machine.

When you are looking at data from the Device Manager database, one of the considerations is whether that information is current. Remember, when you select Physical View for these storage systems there is no active communication with the storage systems. Users could have used all sorts of interfaces to update a storage system configuration, and if it is something that is outside of the purview of Device Manager, then those changes may not be reflected in the Device Manager database.

This is the main reason why it is suggested that you use Device Manager software for all modifications, since it will always be aware of all updates. Whenever you are unsure, you need to do a refresh of the storage system(s). This will synchronize the hardware configuration with the Device Manager database.

The Refresh button at the bottom of the Navigation tree only updates the display. You need to go to the storage system view of a particular unit and click the “Refresh” that is near the middle and top of the display to start the function we are talking about.

The physical view for the Universal Storage Platform and Network Storage Controller does not use the data contained in the Device Manager database. Instead, Storage Navigator is used to provide that function. So, when you launch the view for one of those storage systems, Device Manager software will open an initial window and then redirect the client to communicate with the Service Processor (SVP) to fill in the window with a Storage Navigator screen. The data that gets populated here is from the actual storage system configuration, not the Device Manager database. Please note: if the Device Manager Client and the storage system SVP can not communicate with one another, this request will not work.

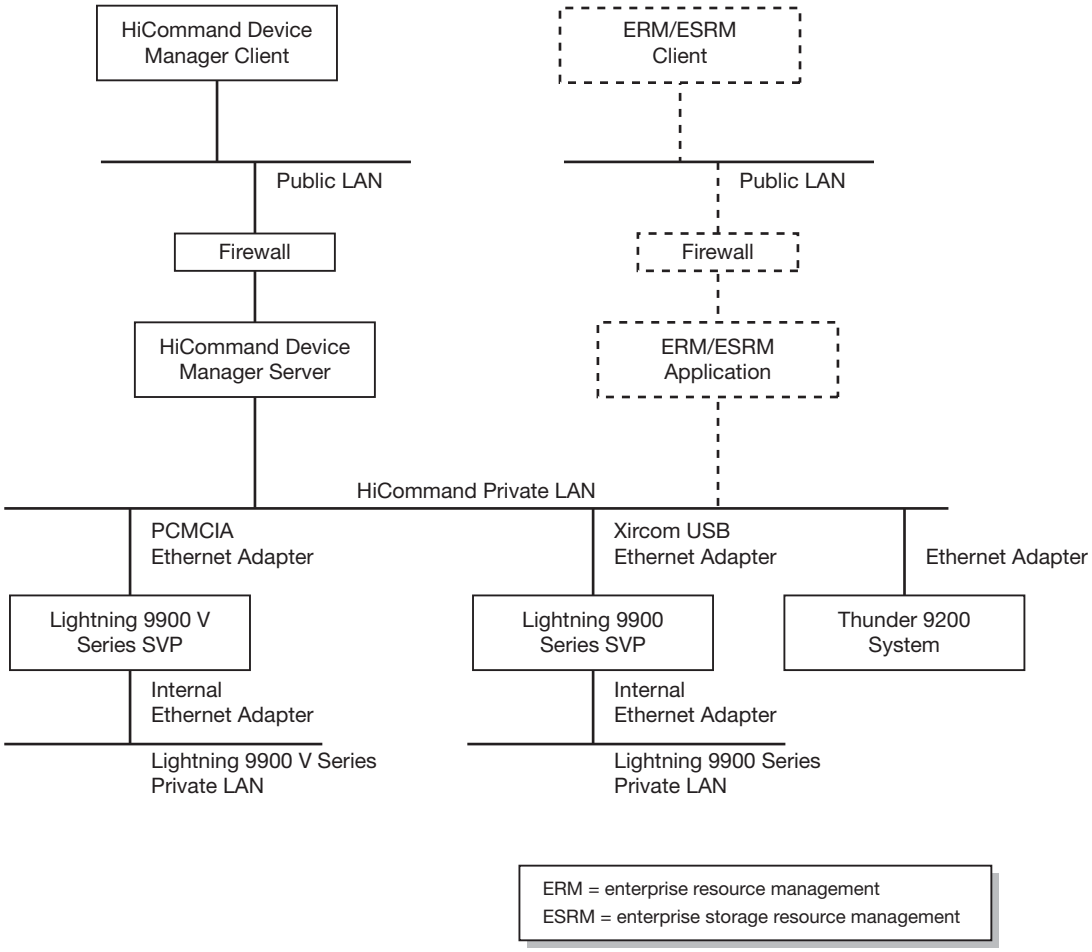
When you launch the Physical View of the Adaptable Modular Storage or Workgroup Modular Storage, Device Manager software will start Storage Navigator Modular (SNM) Web GUI. Though made a little easier with Device Manager 5.0 (you no longer have to create an array unit name with SNM), this unfortunately requires that you to perform some setup so the launch will work properly, such as installing Storage Navigator Modular Web GUI on the same machine as the Device Manager Server. This configuration will not have the same problem as the Universal Storage Platform or Network Storage Controller, if the Adaptable Modular Storage or Workgroup Modular Storage is on a different network. The Device Manager Client will always be in communication with the code on the Device Manager Server machine, either Device Manager or SNM.

Security

Device Manager software can be deployed securely in a variety of operational environments, including private LAN, virtual private network (VPN), corporate intranet, and even public Internet. If needed, many of the security features for Device Manager communications (for example, between Web Client and Server) can be customized for your operational environment.

Device Manager software provides a Server Security overview in the Server Installation Guide, which outlines various scenarios for properly securing the product in your environment. The overview describes in detail all the security options of the product. (See Figure 3 for an overview diagram.)

Figure 3: Server Security with Device Manager Software



Device Manager software provides an end-to-end security context between the user and Device Manager Server, which prevents message tampering and provides privacy, user authentication, and no repudiation.

Any XML client (GUI and CLI) will talk to the Device Manager Server via TCP port 2001 for clear text communications, and port 2443 for XML over HTTP tunneled through SSL. The server side ports are configurable, through the properties files, and client side ports are random, as with most TCP/IP applications.

Communications between XML clients and the server are HTTP-based, and internal communications to the HIRDB database server utilize JDBC. Downstream communications to the managed resources utilize various protocols, such as Java RMI, SNMP, DAMP API, and HTTP. All downstream protocols used fixed ports, which are not configurable. Communications between the Device Manager Server and the Host Agents is done via XML over HTTP. SSL is not available at this time for Host Agent to server communications.

Device Manager software provides an end-to-end security context between the user and Device Manager Server, which prevents message tampering and provides privacy, user authentication, and no repudiation. Messages between the Device Manager Web Client and Device Manager Server are secured when the server is running in security mode 2 (SSL/TSL). The HiCommand Suite Common Component single sign-on feature maintains user authentication across HiCommand products. Device Manager software also provides an audit log of all operations and allows you to combine and archive the audit logs for historical analysis, using the HiCommand Suite Common Component integrated logging feature.

More on Protocols

In a normal Device Manager configuration, data is flying in all sorts of directions and various types of protocols are to contain that data. The client and Host Agent will talk to the Device Manager Server. The Server will also communicate with all managed storage systems. In addition, “conversations” can take place between the Device Manager Server and the vendor client (third-party products that interface with Device Manager software). There is even talk between functions on Device Manager Server (Device Manager and HBase). A lot of this talking will be done in different languages.

Device Manager Client and Device Manager Server—HTTP/XML

Device Manager Host Agent and Device Manager Server—HTTP/XML

Device Manager Server and Lightning 9900 Series system—SNMP/FTP

All communications with the Lightning 9900 Series system are done with SNMP except for the initial discovery of the storage system and a storage system refresh (which internally is an initial discovery). For those functions, Device Manager uses FTP to speed up the transfer. If FTP is not set up correctly at the Lightning 9900 Series system, then those features will revert to using SNMP. You want to avoid this since the task will take an extremely long time to complete with SNMP. Also, if SNMP is not set up correctly at the storage system Device Manager software will not be able to manage the storage system at all.

If another application is running as an SNMP Manager on the Device Manager Server, you will have port conflicts (161) if you are managing a Lightning 9900 Series system. If another application is listening for SNMP Traps (port 162), then the Device Manager Server will not start at all, unless you turn off that feature in the dispatcher.properties file.

Device Manager Server and 9900V/USP/NSC—RMI/SNMP

The setup of SNMP here is only needed to receive traps/alerts. All communications besides alert notifications are carried out by RMI.

As many have observed, if changes are made to a storage system outside of Device Manager (using Storage Navigator, DAMP, SNM, etc.), those changes are not reflected back to the Device Manager Database, and a manual refresh is required. The exceptions to that rule are the Universal Storage Platform and Network Storage Controller. If Storage Navigator is launched from Device Manager and the parameter

server.configchange.enabled = true

in the server.properties file is set, then a refresh will automatically be scheduled as soon as you leave Modify Mode in Storage Navigator.

Device Manager Server and Thunder 9200/Thunder 9500 V Series/Adaptable Modular Storage/Workgroup Modular Storage—DAMP

Device Manager Server and Sun StorEdge T3—HTTP

Device Manager Server and Third-party Clients

- HTTP/XML via the Device Manager XML API or CIM API

As of this writing, most third-party products use the XML API to interface with Device Manager software. There are some products that use both the XML and CIM APIs.

One other thing to consider is that the product will use a large number of ports. In the case of communicating with storage systems, different ports will be employed. (RMI uses 1099, SNMP 161/162, etc.)

Table 1 depicts some of the ports used by the Device Manager Server for Version 3.0 and above (all ports are bi-directional).

Table 1. Ports Used by Device Manager Server

Port	*Process*	*Application*
21	ftp	Device Manager Server (Lightning 9900 Series storage systems)
80	http	Device Manager Server (Sun StorEdge T3 storage systems)
161	snmp	Device Manager Server
162	snmp	Device Manager Server (SNMP Traps)
1099	rmi	Device Manager Server (Lightning 9900 V Series systems)
2000	damp	DAMP API
2001	hicommand	Device Manager Server
2443	hicommand	Device Manager Server (SSL)
3050	ibserver	InterBase
3060	interserver	InterClient
23012	hiscan	Device Manager Host Agent
23015	httpsd	HiCommand Common Component
23016	httpsd	HiCommand Common Component (SSL)
23017	hcmdssvmng	HiCommand Single Signon
23018	hcmdssvmng	HiCommand Single Signon

There are also dynamically assigned ports, which are used for internal communications on the Device Manager Server.

This information is also included in the Device Manager Installation and Configuration manual, section 5.4 “Ports Used by the HiCommand Suite Common Component.”

Third-party Products and Device Manager Software

Hitachi, Ltd., and Hitachi Data Systems are the only vendors that create code that talks directly to Hitachi storage systems. In order for other vendors to query or manage a Hitachi storage system, they need to interface with Device Manager software using either an API or CLI interface.

Two APIs are supported by Device Manager software for such third-party ISV integration. The standard interface is the SNIA-conformant SMI-S CIM API. Documentation for this is available on the www.hds.com Web site, and there are no licensing agreements needed to use the API. There is also a proprietary Device Manager XML API package that is kept “Hitachi Data Systems Confidential” but is available under a special license agreement through the Hitachi Data Systems developer program.

The Device Manager XML API has access to more functionality than what is available in the SNIA SMI-S CIM API. For example, this additional functionality found in the XML API includes:

- Replication Management
- LUSE
- External Storage Information
- Virtual Partition Manager (Universal Storage Platform and Network Storage Controller)

The disadvantage of using the Device Manager XML API is that it can change from release to release, and an ISV partner using it would have to expect rework with some Device Manager releases.

So, although Hitachi Data Systems continues to make the Device Manager XML API available to partners under NDA, partners are encouraged to use the SMI-S API wherever possible.

Vendors who have integrated with Device Manager software using one of the APIs or the CLI include:

- HP (APPIQ)
- Tivoli (IBM)
- Veritas
- EMC
- Softek
- Computer Associates
- Invio
- Storability
- BMC

Hitachi Data Systems could be considered one of those “other vendors” since HiCommand Storage Services Manager software does need Device Manager software in order to manage Hitachi hardware.

Forum.hds.com

One great resource to be aware of is the Hitachi Data Systems Storage Forums Web site. Visit it at:

http://forums.hds.com

The site hosts a User Forum, an open discussion forum (also known as a bulletin board, list group, or discussion board) that is publicly available for anyone to read. On the Forum you can ask questions and compare experiences with peers experienced with HiCommand Device Manager software. In order to participate in the Forum, you must first agree to the Hitachi Data Systems Forum Policy, and then register a username and password. After the registration is complete, you must log in with your username and password in order to add/reply to topics.

Best Practices

Database Backup and Export Operations

Introduction

The following describes Best Practice scenarios for backing up and restoring HiCommand databases. This document is written for Windows; however, the principles apply to other platforms, such as Linux and UNIX.

Services

Device Manager 5.x relies on a minimum of four services. They are listed below. The services must be started in the order shown. If they are stopped, they must be stopped in reverse order.

- HiRDB/Embedded Edition _HD0 *
- HiCommand Common Web Service
- HiCommand Single Sign On Service
- HiCommandServer

Notes(*): HiRDB is actually stopped and started via BAT files which are kept in this directory:

C:\Program Files\HiCommand\Base\bin

Users should never stop and start this service via the “Services” option in Windows.

The following documentation describes backup, export, and other procedures. Specimen BAT files and UNIX shell scripts are available to perform these functions and they can be downloaded from the same site as this documentation.

For the rest of this document, “Common Web Service” is abbreviated to CWS and “Single Sign On Service” is abbreviated to SSO.

Backup/Restore

Like any database, the HiCommand databases should be backed up daily. The suggested Best Practice is to execute a script to, as minimum, back up the Common Component database and the Device Manager database. If HiCommand Replication Monitor or HiCommand Tiered Storage Manager software are installed, their databases should be backed up as well.

Backup: Execution of the backup script requires that HiCommandServer is stopped. The specimen script does the following:

- Stops HiCommandServer
- Stops CWS and SSO
- Checks that HiRDB is still running
- Deletes E:\Device Manager5\oldbackup
- Renames E:\Device Manager5\backup to E:\Device Manager5\oldbackup
- Backs up HiCommand databases to E:\Device Manager5\backup
- Starts CWS and SSO
- Starts HiCommandServer

Backup is a quick operation. There is no reason not to do this daily. The backup file is called backup.hdb. However, the script should also back up the HiCommand config files and directories as well.

Restore: Execution of a restore script requires that all services are stopped. The specimen script does the following:

- Stops HiCommandServer
- Stops HiRDB, SSO, and CWS
- Checks that HiRDB, SSO, and CWS are not running
- Restores HiCommand databases from E:\Device Manager5\backup\database\backup.hdb
- Starts HiRDB, SSO, and CWS
- Starts HiCommandServer

Notes: The HiCommand databases are restored by this operation. The config files that were backed up at the same time are NOT restored. If you have made changes to the Device Manager configuration, such as enabling SSL, this restore will NOT reverse those changes. This may not be what you expected!

You **MUST** restore the HiCommand databases to the same drive/directory that they were backed up from. Thus, you **cannot** do the following:

Back up Device Manager 5.x from C: drive

- Install a new copy of Device Manager 5.x on a new server on D: drive.
- Restore Device Manager 5.x database on the new server.

The backup contains the drive/directory information for the original databases. The restore will fail. If you want to do this, you must use Export/Import. Refer to the next section of this document. However, you can do this:

- Back up Device Manager 5.x from C: drive.
- Install a new copy of Device Manager 5.x on a new server on C: drive.
- Restore Device Manager 5.x database on the new server.

Be warned that:

- The versions of Device Manager 5.x **MUST** be the same. You cannot restore a Device Manager 4.2 database on a Device Manager 5.0 system.
- You must synchronize the Common Component and HiCommandServer databases on the new machine. You do this by editing the file *"server.properties"* in the following directory:

C:\Program Files\HiCommand\DeviceManager\HiCommandServer\config

Change this:

Synchronize Device Manager database and Common Repository in starting Device Manager Server

server.base.initialsynchro=false

To this:

Synchronize Device Manager database and Common Repository in starting Device Manager Server

server.base.initialsynchro=true

Edit this file before you run the restore. Then, the databases will be synchronized when the services are started after the restore has completed.

Change the variable back to “false” when HiCommandServer has finished initialization and you can log on successfully. This is not a requirement. However, if you do not do this, every subsequent stop/start of Device Manager software will be slower as unnecessary synchronization will be performed.

- You must run “Device Manager5 URL.BAT” (see later) if you have restored the Device Manager databases on another server.

Export/Import

Export and Import are the “ultimate” backup and restore procedures. The export can be imported to the same server, another server, or even a later version of Device Manager software. However, Export and Import are slow. This is due to the fact that the data and the database table definitions are exported and imported.

Make sure that you have a good backup of the Device Manager databases before you try the import procedure!

Export: Execution of an export script requires that all services except HiRDB are stopped. The specimen script does the following:

- Stops HiCommandServer
- Stops SSO and CWS
- Checks that HiRDB is still running
- Deletes E:\Device Manager5\oldexport
- Renames E:\Device Manager5\export to E:\Device Manager5\oldexport
- Exports HiCommand databases to E:\Device Manager5\Export
- Starts SSO and CWS
- Starts HiCommandServer

As previously stated, export is a slow operation. It is recommended that you perform this operation weekly. In addition, if you wish to send a copy of the database to another user for testing and/or debugging, always use export, not backup.

Import: Execution of an import script requires that all services except HiRDB are stopped. The specimen script does the following:

- Stops HiCommandServer
- Stops SSO and CWS
- Checks that HiRDB is still running
- Imports HiCommand databases from E:\Device Manager5\Export
- Starts SSO and CWS
- Starts HiCommandServer

Notes: Remember to synchronize the Common Component and Device Manager databases using the procedure described above in the section on Restore.

Remember to run *"Device Manager5 URL.BAT"* (see later) if you have imported the Device Manager databases on another server.

Running Multiple Java Runtime Environments (JREs)

The Java Runtime Environment (JRE) is a Java Interpreter that is needed to run Java applications that do not have an included interpreter. Almost all of the code for the Device Manager Server is designed as a Java Servlet, which does not require an installed JRE at the client. The functions that are the exceptions are Resource Group Management and the Physical View of certain storage systems. Those storage systems include the Thunder 9200 system, the Thunder 9500 V Series systems, the Lightning 9900 Series systems, and the Lightning 9900 V Series systems. So, if you wish to use those functions you must install an appropriate JRE on the Client machine.

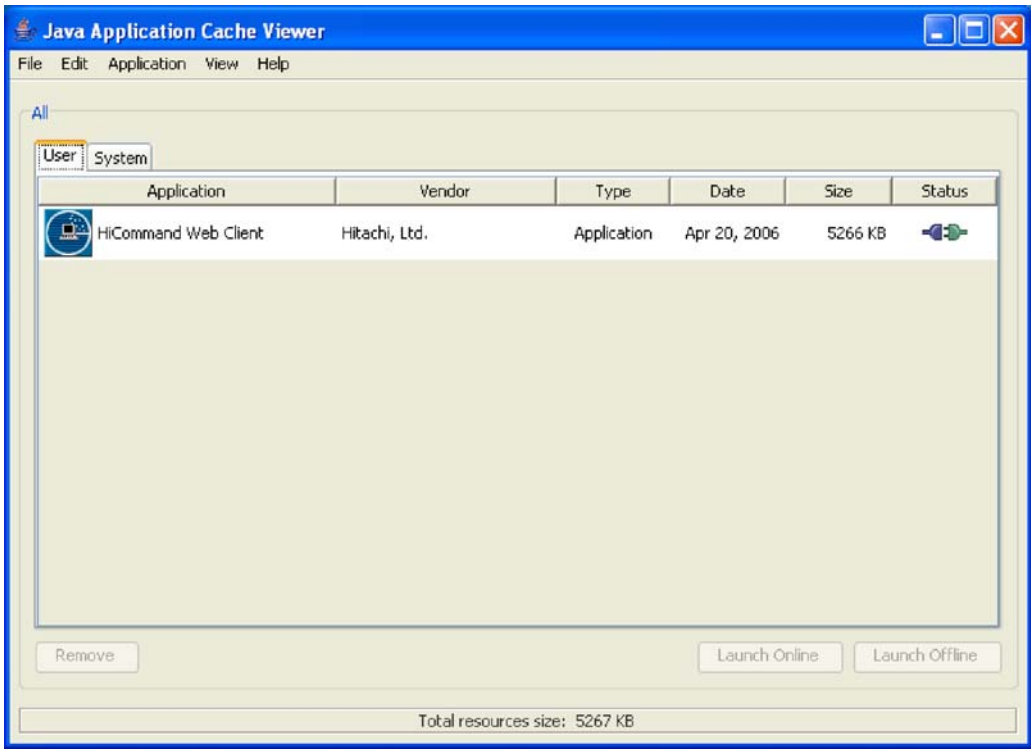
There is no requirement for the customer to install a JRE on a Device Manager Server or a host running a Device Manager Host Agent (unless you wish for that machine to operate as an Device Manager Client). When you install the Device Manager Server code it will "silently" install the necessary JAVA code within its own folder structure. The word "silent" is used here to designate that this code is available only for the Device Manager Server, and its existence is not made known globally to the other applications on that platform.

This is also true of the Host Agent, at least at the Version 5 level. Earlier versions of the software required the person installing the product to find the correct JRE on the Internet for each platform. With Version 5 that is not necessary, unless you are going to run the Device Manager Client on that server.

If the user needs to install multiple JREs, due to other product requirements, you need to use the Java Web Start (JWS) to properly set up the environment so the Device Manager Client will use the correct JRE. Many users are installing JRE 1.5, but we do not support that version as of this writing. If you have 1.5 installed, please use the following steps to allow it to coexist with a supported version of the JRE for Device Manager software.

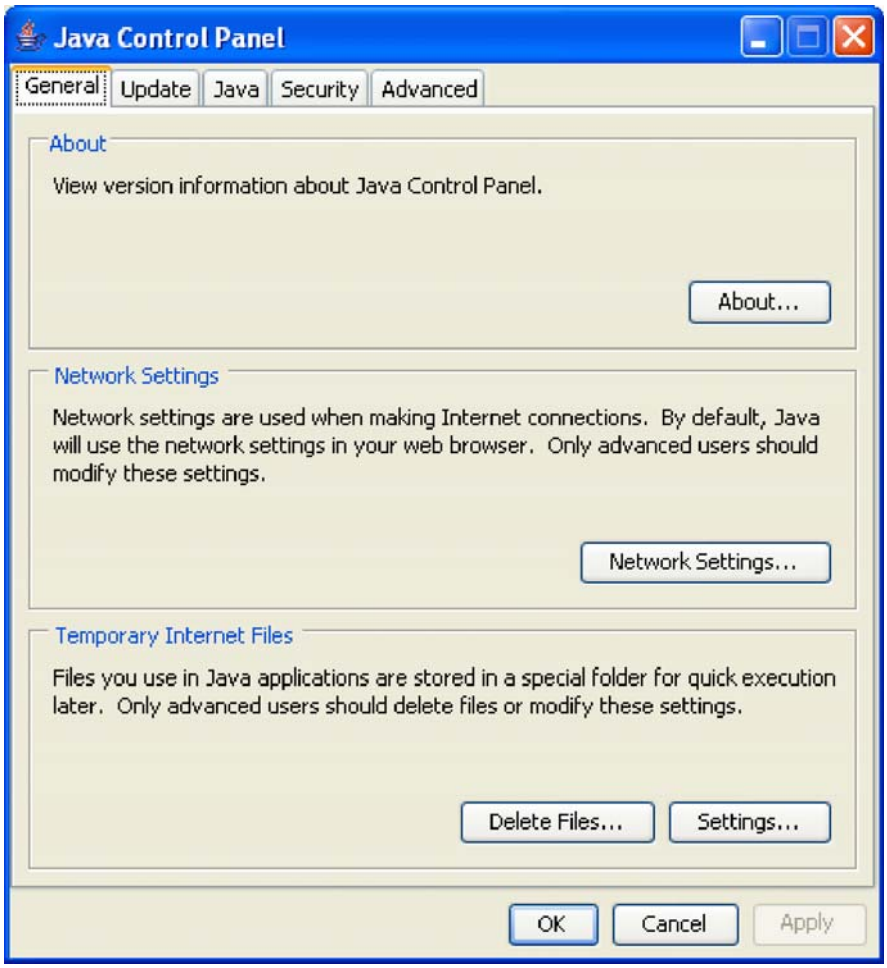
1. Open JWS. (See Figure 4. If you cannot find Open JWS, you may have to search for javaws, and select the one that is in the 1.5 directory.)

Figure 4. Java Application Cache Viewer



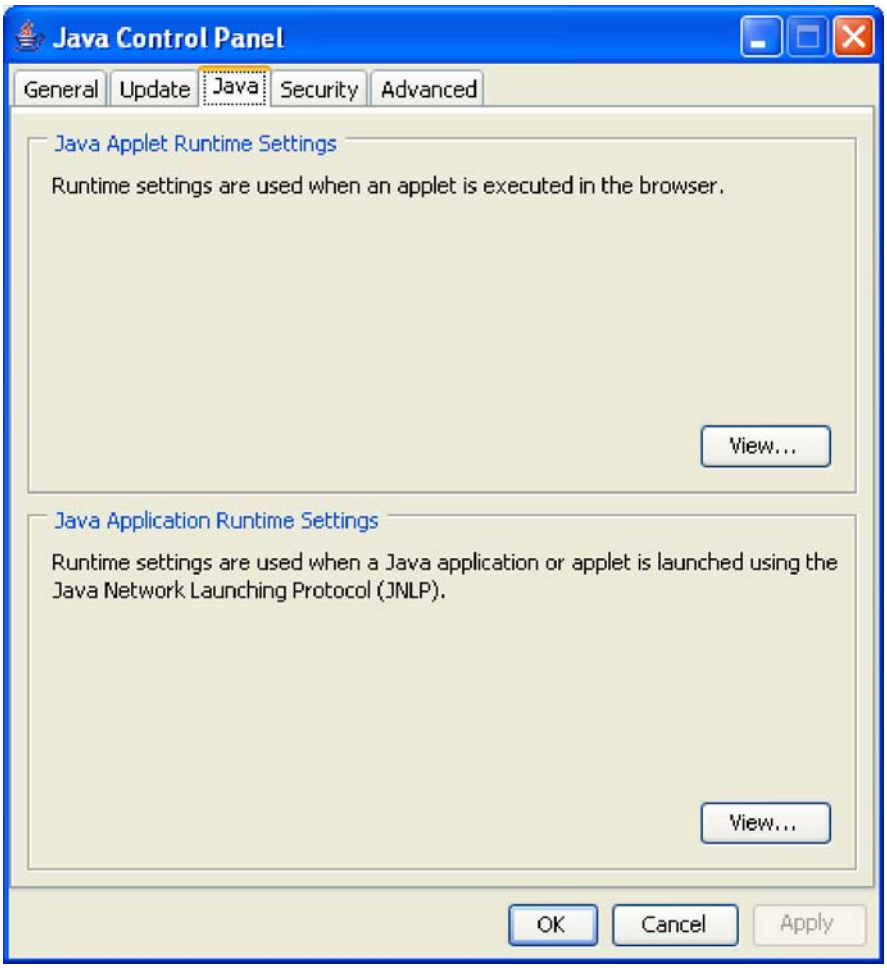
2. On the menu, select Edit, Preferences and you will get the Java Control Panel (see Figure 5).

Figure 5. Java Control Panel



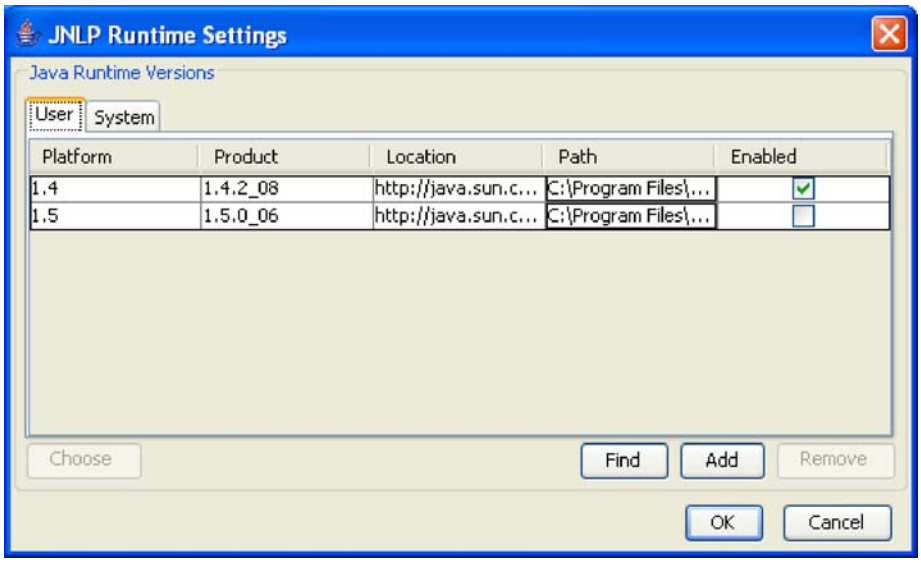
3. Next, click on the Java tab and select the Java Application Runtime Settings at the bottom of the screen (see Figure 6).

Figure 6. Java Control Panel—Runtime Settings View Options



4. On the final screen, make sure that the supported version of the JRE is Enabled and all the others are not.

Figure 7. JNLP Runtime Settings



5. Once this is done you should be able to launch both Resource Group Management and supported Physical Views.

Replacing HBAs

Here is a Best Practice on how to change WWN configurations on a Host or how to replace HBAs attached to a Host. It applies to cases where the user needs to replace HBAs on hosts and wants this to be reflected on Device Manager Host definitions.

It describes how you can accomplish this by GUI.

- Identify host you want to replace HBA
  - In this case, take Lab_test_00 as an example.
  - One port HBA with a WWN of **11.22.33.AA.44.55.AA.88** is now installed on this host.
  - Assuming you want to replace this HBA with another one with a WWN of **10.00.00.00.C9.29.50.70**.

    Steps for keeping access LUNs assigned to the host from OS are:

    1. Add new WWNs to the host. (Make it sure the host can be accessed by assigned LUNs from new HBAs that have new WWNs.)
    2. Replace HBA physically and remember the OS may need to be rebooted.
    3. LUNs assigned for the host can now be accessed through new WWNs. (You may need to reconfigure your multipath software, such as Dynamic Link Manager software.)
    4 Remove old WWNs from the host.

       Both (1) and (4) are steps that you can do on Device Manager GUI or CLI.

    To do step 1, assigning new WWNs to the host:

    1. Specify host you want to change HBA.

2. Click Modify Properties and you will have another window to see the current WWN configuration.
3. Specify WWNs on HBAs that will be newly installed.
4. Now there are two WWNs (old and new) defined on this host object. Click OK to go to next screen.
5. This screen asks you which LUN you want to modify. In this case, click “select all” then OK to go next screen.
6 Complete to add new WWN to target Host.

- At this moment Lab_test_00 has two WWNs.
- That means all LUNs assigned for this host can be accessed from both WWN port.
- Stop the host and replace the old HBA (11.22.33.AA.44.55.AA.88) with the new one (10.00.00.00.C9.29.50.70), and then reboot the host.
- The new HBA WWN has been configured on storage system and the host can access same LUNs using newly installed HBA.
  1. Specify host you want to change HBA.
  2. Click Modify Properties and you will have another window to see the current WWN configuration.
  3. Specify WWNs on old HBA and click Remove then OK.
  4. Complete to delete old WWN from the target Host.
  - Now the target host, Lab_test_00, has newly installed HBA WWNs. (10.00.00.00.C9.29.50.70)
  - All LUNs associated with the host are now secured (LUN secured) by new WWNs.

Support—Gathering Logs for the Technical Response Center (TRC)

A number of things should be collected whenever there is a problem and a case has been opened with the TRC. Before discussing log collection it is very helpful if the following information is included in the documentation gathering:

- Model, speed, and memory of Device Manager Server
- Screen shots of any errors or applicable windows
- Date and time of the error
- Time difference between the Device Manager Server and the involved storage system(s)

Device Manager produces a large amount of log data and sometimes it can be overwhelming. Therefore, if it is possible to recreate the error, a step that would help greatly in identifying what is wrong is to limit the amount of data in the logs. To do that before you recreate the problem, stop the Device Manager services/processes and move all the logs (files in the <install location>/DeviceManager/HiCommandServer/logs) to another location for safekeeping. When you start up all the services/processes all new logs will be created. Then, when the recreation test is done, the logs will have data that is mostly related to the problem.

As of this writing, the latest versions of the TRC getconfigs for Windows and UNIX have been updated to run the log collection routines for both a Host Agent or server, if they are running on the affected platform. While this will cover all the needed data in most cases, some customers may not want to download the getconfigs. Following are the procedures for gathering the Device Manager logs (in all cases we are assuming the install was done to the default locations).

Device Manager Server

Windows – C:\Program Files\HiCommand\Base\bin\hcmdsgetlogs /dir <name of an empty or non-existent folder> /type DeviceManager

Solaris - /opt/HiCommand/Base/bin/hcmdsgetlogs –dir <name of an empty or non-existent folder> -type DeviceManager

This command will create 4 Java Archive files (type JAR). These files need to be uploaded to the TRC FTP site (**https://tuf.hds.com**). These files can be opened with WinZip.

Device Manager Host Agent

Windows - C:\Program Files\Hitachi\HDVM\agent\bin\tic

UNIX (except AIX) - /opt/HDVM/HBaseAgent/bin/tic

AIX - /usr/HDVM/HBaseAgent/bin/tic

The output will be all the necessary log files stored in a folder called resultDir. The contents of this directory should be compressed and uploaded to TUF.

If the problem is with a third-party vendor product, an additional step may be necessary. To get more of what is going on in the database one of the logger parameters will need to be changed. The setting is logger.hicommandbase.loglevel=, which can be found in the logger.properties file. By default this is set to 20. This should be set to 30 to help investigate a problem with a third-party vendor product. In order for this change to be picked up you will need to restart the Device Manager Server.

Hitachi Data Systems Corporation

Corporate Headquarters 750 Central Expressway, Santa Clara, California 95050-2627 USA
Contact Information: 1 408 970 1000 www.hds.com / info@hds.com

Asia Pacific and Americas 750 Central Expressway, Santa Clara, California 95050-2627 USA
Contact Information: 1 408 970 1000 info@hds.com

Europe Headquarters Sefton Park, Stoke Poges, Buckinghamshire SL2 4HD United Kingdom
Contact Information: + 44 (0) 1753 618000 info.uk@hds.com

WHP-249-00 March 2007