



# Intel® Carrier Grade Server Provisioning Development Kit

User Guide

---

*Revision 1.1*

## Legal Statements

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS FOR THE GENERAL PURPOSE OF SUPPORTING INTEL DEVELOPED SERVER BOARDS AND SYSTEMS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR GENERAL PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS OTHERWISE AGREED IN WRITING BY INTEL, THE INTEL PRODUCTS ARE NOT DESIGNED NOR INTENDED FOR ANY APPLICATION IN WHICH THE FAILURE OF THE INTEL PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Intel® Active Management Technology requires the computer system to have an Intel AMT-enabled chipset, network hardware and software, as well as connection with a power source and a corporate network connection. With regard to notebooks, Intel AMT may not be available or certain capabilities may be limited over a host OS-based VPN or when connecting wirelessly, on battery power, sleeping, hibernating or powered off. For more information, see <http://www.intel.com/technology/manage/iamt/>

Any software source code reprinted in this document is furnished under a software license and may only be used or copied in accordance with the terms of that license. Refer to the license corresponding to the software component provided in the development kit for details. Portions of this development kit may be supplied under Open Source License agreements.

Intel, Intel Core, and Xeon are trademarks of Intel Corporation in the U.S. and other countries.

\* Other names and brands may be claimed as the property of others.

Copyright © 2007 Intel Corporation. All rights reserved.

Portions Copyright © 2005 LSI Logic Corporation. All rights reserved

## Revision History

Date	Revision	Description
11/27/2007	1.1	Initial release.

# Contents

---

Legal Statements .....	ii
Revision History .....	ii
<b>Contents .....</b>	<b>iii</b>
<b>1 Introduction .....</b>	<b>1</b>
What's in the Development Kit? .....	1
Scripting Tools .....	1
Platform Tools .....	2
Source Code .....	2
Which Solution to Choose? .....	2
Supported Intel® Products .....	4
Related Documents .....	4
Tool Selection Guide .....	5
Drivers .....	7
Required Drivers .....	7
How To Load Drivers .....	8
Sample Script Flow .....	8
Installation .....	9
Runtime Environment .....	10
Set Deployment Environment Variables (setDeploymentEnv) .....	10
List of Environment Variables .....	10
Prepare Deployment Environment (prepareDeployment) .....	11
Customizing a Deployment .....	12
Important Environment Variable Behavior .....	13
Tools for Setting Environment Variables in Windows .....	13
<b>2 Utilities .....</b>	<b>15</b>
AMTCfg .....	15
Supported Intel® Platforms .....	15
Supported Environments .....	15
Command Line Syntax .....	15
Return Codes .....	16
Script Examples .....	16
CmdTool2 .....	17
Supported Intel® Platforms .....	17
Supported Environments .....	17
Dependencies .....	17
Command Line Syntax .....	18
Script Examples .....	18
Disc-Wipe .....	19

Supported Platforms.....	19
Supported Environments.....	19
Command Line Syntax.....	19
Script Examples.....	19
Erase .....	20
Supported Intel® Platforms.....	20
Supported Environments.....	20
Command Line Syntax.....	20
Return Codes .....	20
Script Examples.....	21
dmidecode.....	22
Supported Intel® Platforms.....	22
Supported Environments.....	22
Command Line Syntax.....	22
ETHTOOL .....	23
Supported Intel® Platforms.....	23
Supported Environments.....	23
Command Line Syntax.....	23
Script Examples.....	23
Additional Documentation .....	23
FRUSDR .....	24
Supported Intel® Platforms.....	24
Supported Environments.....	24
Command Line Syntax.....	24
Return Codes and Error Messages .....	25
Script Examples.....	30
FWPIAUPD.....	31
Supported Intel® Platforms.....	31
Supported Environments.....	31
Command Line Syntax.....	31
BMC Updates .....	31
HSC updates.....	32
Intel LCP Updates .....	34
Error Messages and Return Codes .....	35
Script Examples.....	38
hdparm.....	39
Supported Intel® Platforms.....	39
Supported Environments.....	39
Command Line Syntax.....	39
Return Codes .....	41
Script Examples.....	41
Additional Documentation .....	41

HyperCfg and Spy.....	42
Supported Intel® Platforms.....	42
Supported Environments.....	42
Command Line Syntax.....	42
Return Codes.....	44
Script Examples.....	46
Hyperwin and Spy.....	47
Supported Intel® Platforms.....	47
Supported Environments.....	47
Command Line Syntax.....	47
Return Codes.....	49
Script Examples.....	51
persistVariable.....	52
Supported Intel® Platforms.....	52
Supported Environments.....	52
Command Line Syntax.....	52
Return Codes.....	52
Script Examples.....	53
iFlash32.....	54
Supported Intel® Platforms.....	54
Supported Environments.....	54
Command Line Syntax.....	54
Return Codes.....	55
Script Examples.....	55
Additional Documentation.....	55
Intel® One-Boot Flash Update.....	56
Supported Intel® Platforms.....	56
Supported Environments.....	56
Command Line Syntax.....	56
Return Codes.....	57
Script Examples.....	57
Usage Notes.....	57
Additional Documentation.....	58
IpmiUtil.....	59
Supported Intel® Platforms.....	59
Supported Environments.....	59
Command Line Syntax.....	59
Return Codes.....	60
Script Examples.....	60
Additional Documentation.....	60
Log Manager Utility.....	61
Supported Intel® Platforms.....	61

Supported Environments .....	61
Command Line Syntax .....	61
Return Codes .....	61
Script Examples .....	62
nicSpeed .....	63
Supported Intel® Platforms .....	63
Supported Environments .....	63
Command Line Syntax .....	63
Return Codes .....	63
Script Examples .....	63
RaidCfg32 .....	64
Supported Intel® Platforms .....	64
Supported Environments .....	64
Command Line Syntax .....	64
Reboot .....	66
Supported Intel® Platforms .....	66
Supported Environments .....	66
Command Line Syntax .....	66
Return Codes .....	66
Script Example .....	66
SCSI RAS Tools for Linux .....	68
Supported Intel® Platforms .....	68
Supported Environments .....	68
Command Line Syntax .....	68
Script Examples .....	68
Additional Documentation .....	68
SCSI RAS Tools for Windows .....	69
Supported Intel® Platforms .....	69
Supported Environments .....	69
Command Line Syntax .....	69
SLIST Syntax .....	69
SDISKFW .....	70
SDISKMON .....	70
Syscfg .....	72
Supported Intel® Platforms .....	72
Supported Environments .....	72
Command Line Syntax .....	72
Script Examples .....	72
Additional Documentation .....	72
System Information Manager .....	73
Supported Intel® Platforms .....	73
Supported Environments .....	73

Command Line Syntax .....	73
Source, Attributes, and Name Parameters .....	74
Return Codes .....	78
Script Examples.....	79
<b>Glossary .....</b>	<b>81</b>
<b>Index.....</b>	<b>83</b>





# 1 Introduction

---

The Intel® Carrier Grade Server Provisioning Development Kit provides tools for writing scripts that install, configure, and update Intel servers.

Typical applications include

- An equipment vendor creating a system deployment DVD
- A manufacturer automating system integration
- An IT professional automating deployment and system updates
- An IT professional creating a USB flash-drive for on-site work

The tools execute on the target machine, either in-service with an Operating System installed, or out-of-service with a temporary bootable media. Scripts can operate with or without a network connection. Local or remote keyboard and video are not required.

The scripting tools enables you to write common scripts that can be shared across multiple products, if desired. Unattended scripts can detect the underlying hardware and make decisions about which operations to perform.

Scripts can load firmware, create RAID arrays, enable LAN channels, install an Operating System, and load drivers and applications without human intervention. If multiple reboots are required, the scripting tools can track the install phase in the platform's nonvolatile storage.

Since deployment and maintenance models vary widely, this development kit does not attempt to lock the user into any particular model. Instead, the development kit is designed to operate within existing deployment frameworks across various topologies without forcing a specific deployment model upon the user. Network capability is available but not required.

## What's in the Development Kit?

The development kit bundles intelligent scripting tools and platform tools into a single development kit supporting multiple hardware products.

This User's Guide describes the components of the development kit and how they can be used in various deployment activities. Individual tools may also be documented in separate documents provided with the development kit.

## Scripting Tools

The scripting facility provides tools for the following operations:

**Component Detection.** After taking an inventory of platform hardware and software, scripts can query the inventory and take appropriate actions.

**Deployment Logging.** As deployment tasks progress, scripts can capture tool results and place log entries in a central file. Scripts can also redirect “stdout” and “stderr” if desired.

**Reboot and Resume.** Deployment procedures often require multiple reboots. Scripts can issue an Operating System or hardware reboot (to a new boot device, if required), and track their progress in nonvolatile storage.

**System Erase.** An erase utility restores the machine to a default factory state, erasing disk drives, removing non-volatile data, and restoring default configuration values.

## Platform Tools

Platform tools vary by hardware platform but generally include the following capabilities:

- Upgrade and configure BIOS
- Upgrade and configure management firmware
- Create and configure RAID arrays
- Configure LAN channels

In addition, you can modify the baseboard BIOS image to include a custom splash screen, user binary code, or custom defaults. The development kit provides tools for modifying the BIOS flash file prior to installation on a machine. (These tools are intended for engineering or factory use and not for use in the field.)

## Source Code

The following list shows the location of the source code for the tools and drivers distributed with this tool kit under an Open Source license.

IPMIUtil	<a href="http://ipmiutil.sf.net">http://ipmiutil.sf.net</a>
NICspeed Configuration VBScript	vbs source in Intel Carrier Grade Server PDK
Ethtool	<a href="http://sourceforge.net/projects/gkernel/">http://sourceforge.net/projects/gkernel/</a>
SCSI RAS Tools	<a href="http://scsirastools.sf.net">http://scsirastools.sf.net</a>
hdParm	<a href="http://sf.net/projects/hdparm/">http://sf.net/projects/hdparm/</a>
disc-wipe	<a href="http://sourceforge.net/projects/disc-wipe/">http://sourceforge.net/projects/disc-wipe/</a>
SMBiosDecode**	Source in Intel Carrier Grade Server PDK
Intel IMB driver for IPMI	Source in Intel Carrier Grade Server PDK
SMI driver for syscfg	Source in Intel Carrier Grade Server PDK
Intel e1000 driver	<a href="http://sourceforge.net/projects/e1000">http://sourceforge.net/projects/e1000</a>

\*\*This tool is only called by other scripts in the Intel Carrier Grade Server PDK.

## Which Solution to Choose?

In addition to the Intel Carrier Grade Server Provisioning Development Kit, Intel provides deployment tools in other formats to suit other usage models.

**Intel® Deployment Assistant** provides an easy-to-use graphical tool designed to

speed the deployment of Intel servers. Intel Deployment Assistant helps get a server up and running quickly through a step-by-step wizard that guides technicians through the set-up process. Using this tool, the user can easily update the drivers or firmware, plus configure the most common system settings. System Administrators can also use Intel Deployment Assistant to perform unattended Operating System installations.

**Intel® Rapid Boot Toolkit** is a collection of tools for the administrators and programmers supporting High Performance Computing (HPC) clusters of certain Intel server boards. With this toolkit, you can boot the HPC nodes using the Intel® Rapid Boot BIOS, and then launch a flash-based payload that supports your HPC provisioning needs. By using the Intel Rapid Boot Toolkit, you can boot your entire HPC cluster faster and then provision it in a fraction of the time required by the conventional BIOS and PXE provisioning method. The Intel Rapid Boot Toolkit and Intel Rapid Boot BIOS are supported on select Intel servers.

Here are some general guidelines on which tool kit to use:

- Use the Intel Deployment Assistant to interactively perform occasional activities of a single machine using a menu-based graphical user interface.
- Use the Intel Rapid Boot Toolkit to perform automated, repeatable operations on a cluster of servers using a network-based scheduler, particularly if the Intel® Rapid Boot BIOS is present.
- Use the Intel Carrier Grade Server Provisioning Development Kit to create automated, repeatable scripts that can be shared across multiple machine types and without requiring connection to a network.

Table 1 describes the differences in more detail.

**Table 1. Deployment Models**

	<b>Intel® Deployment Assistant</b>	<b>Intel® Carrier Grade Server Provisioning Development Kit</b>	<b>Intel® Rapid Boot Toolkit</b>
<b>Deployment Model</b>	Interactive GUI	Unattended Scripts	Unattended Network Job Scheduler
<b>Typical Users</b>	Customer IT Department, VAR, System Integrators	Factory OEM/TEM/VAR Customer IT Department	Factory OEM/TEM/VAR Customer IT Department
<b>Portability</b>	One CD per product	One development kit for multiple products	One toolkit for multiple products
<b>Mass Deployment</b>	One server at a time	Servers of different types can call the same script simultaneously.	Each server retrieves tasks from a network job queue.
<b>Network</b>	Runs locally from local/remote CD image	Runs locally from local/remote file system	Runs locally from remote file system and job queue
<b>Data Sources</b>	CD USB Network Hard Disk	CD/DVD USB Hard Disk Network	Network

	Intel® Deployment Assistant	Intel® Carrier Grade Server Provisioning Development Kit	Intel® Rapid Boot Toolkit
<b>Operating System Support</b>	Windows, Linux	Windows, Linux	Linux
<b>Delivered as</b>	Bootable CD Image	ZIP file	TAR file

## Supported Intel® Products

This development kit provides tools for the following Intel® products:

- Intel Server Platform SR1530AH
- Intel Server Platform SR2500AL
- Intel IP Network Server NSW1U
- Intel IP Network Server NSC2U
- Intel Carrier Grade Server TIGW1U
- Intel Carrier Grade Server TIGH2U

The development kit contains tools that operate under the following versions of Linux\* and Windows\*:

- Microsoft Windows Preinstallation Environment 2005, 32-bit x86 Edition
- Microsoft Windows Server 2003, 32-bit x86 Edition
- Red Hat Enterprise Linux 4, Update 4, 32-bit BOOT kernel
- Red Hat Enterprise Linux 4, Update 4, 32-bit ELsmp kernel

## Related Documents

You can learn more about Intel's deployment tools in the following documents.

**Table 2. Related Documents**

Title	Number	Location
Intel® Rapid Boot Toolkit	<a href="#">D96629</a>	<a href="http://developer.intel.com/">http://developer.intel.com/</a>
Intel® Deployment Assistant	<a href="#">311579</a>	<a href="http://developer.intel.com/">http://developer.intel.com/</a>
Intel® One-Boot Flash Update Utility User Guide	N/A	TIGW1U Deployment CD
Intel® System Configuration Utility User Guide	E12458-005	TIGW1U Deployment CD
Intel® Remote Management Module 2	<a href="#">311580</a>	<a href="http://developer.intel.com/">http://developer.intel.com/</a>
Intel® Matrix Storage Manager Quick Start Guide	<a href="#">CS-020659</a>	<a href="http://developer.intel.com/">http://developer.intel.com/</a>
Intel® Matrix Storage Manager User's Manual	<a href="#">CS-020670</a>	<a href="http://developer.intel.com/">http://developer.intel.com/</a>
Intel® Embedded Server RAID Technology II, Intel® Integrated Server RAID, and Intel® RAID Controllers	<a href="#">D29305-004</a>	<a href="http://developer.intel.com/">http://developer.intel.com/</a>

Title	Number	Location
SRCSAS18E and SRCSAS144E Software User's Guide		
Intel® Server Boards S3000AHLX, S3000AH, and S3000AHV Technical Product Specification	<a href="#">D72579-002</a>	<a href="http://developer.intel.com/">http://developer.intel.com/</a>
Intel® Server Board S5000PAL / S5000XAL Technical Product Specification	<a href="#">D31979-006</a>	<a href="http://developer.intel.com/">http://developer.intel.com/</a>
Intel® IP Network Server NSW1U Technical Product Specification	<a href="#">D55559-003</a>	<a href="http://developer.intel.com/">http://developer.intel.com/</a>
Intel® Carrier Grade Server TIGW1U Technical Product Specification	<a href="#">D82753-002</a>	<a href="http://developer.intel.com/">http://developer.intel.com/</a>
Intel® IP Network Server NSC2U Technical Product Specification	N/A	<a href="http://www.intel.com/support/telecom/computeboards/nsc2u/index.htm">http://www.intel.com/support/telecom/computeboards/nsc2u/index.htm</a>
Intel® Carrier Grade Server TIGH2U Technical Product Specification	N/A	See your local Intel field representative

## Tool Selection Guide

For most platform components, this development kit provides more than one tool that can be used to read, modify, or update the component. Check the operating system support for each tool to determine if it meets your requirements.

**Table 3. Tool Selection Guide**

Platform Component	Tool	Sample Scripts
BIOS updates	Intel OFU IFlash32 (Microsoft Windows Preinstallation Environment 2005 only)	flash-s5.sh or flashbios-s3.sh flash-s5.cmd or flashbios-s3.cmd iflash.py, ofu.py, flash.py
BIOS settings	syscfg	cfgbios.sh, cfgbios.cmd
BMC updates	FWPIAUPD (Microsoft Windows Preinstallation Environment 2005 only) Intel OFU	flash-s5.cmd flash-s5.sh fwpiaupdt.py, ofu.py, flash.py
BMC settings	IPMIUtil	cfgbmc.sh, cfgbmc.cmd
FRU/SDR	Intel OFU FRUSDR (Microsoft Windows Preinstallation Environment 2005 only) IPMIUtil (selected FRU data only)	flash-s5.sh flash-s5.cmd cfgbmc.sh, cfgbmc.cmd frusdr.py
Hot-Swap Controller (HSC) updates	Intel OFU FWPIAUPD (Microsoft Windows Preinstallation Environment 2005 only)	flash-s5.sh flash-s5.cmd
NIC Settings	NIC Configuration VBScript Ethtool	nicSpeed.vbs nicSpeed.cmd cfgnic.sh cfgnic.cmd
Hard disk drives and RAID	SCSI RAS Tools CMDTool2 RAIDcfg Hypercfg Spy hdParm	cfgdisk-scsi.sh, cfgdisk-scsi.cmd cmdtool2-createvolume.sh, cmdtool2- createvolume.cmd raidcfg-createvolume.cmd hypercfg-createvolume.sh cfgdisk-ata.sh, cfgdisk-ata.cmd
Intel Remote Management Module 2	KiraTool (available separately with Intel RMM2)	rmm-info.cmd
System Reboot	Reboot	reboot-s5.sh, reboot.sh, reboot.cmd
Scripting support	addLogRecord setLogLevel getLogSeverity addProperty queryProperty persistVariable	addLogRecordDemo.sh, addLogRecordDemo.cmd, addLogRecordDemo.py setLogLeveDemo.sh, setLogLeveDemo.cmd, setLogLeveDemo.py getLogSeverityDemo.sh, getLogSeverityDemo.cmd, getLogSeverityDemo.py addPropertyDemo.sh, addPropertyDemo.cmd, addPropertyDemo.py queryPropertyDemo.sh, queryPropertyDemo.cmd, queryPropertyDemo.py persistVariableDemo.sh, persistVariableDemo.cmd
Erase system	Erase disc-wipe	erase.sh, erase.cmd
Intel AMT configuration settings	AMTcfg	flashamt.sh, flashamt.cmd, flashamt.py cfgamt.sh, cfgamt.cmd, amtcfg.py

# Drivers

This section lists the drivers used by the tool kit.

## Required Drivers

The following table shows the drivers required by the tools in the development kit.

**Note:** Most of these drivers could be loaded on the media by default, except for the imb and RAID drivers, which should be dynamically loaded as needed.

**Table 4. Drivers**

Intel® Server Platform Family	Operating System	Tool	Driver	Version	Location of Source
S5000	Linux	Any IPMI	imb.ko	Build 22	GPL, in Intel Carrier Grade Server PDK
S5000	Microsoft Windows Preinstallation Environment 2005	Any IPMI	Imbdrv.sys	v10.0	Binary, Intel
S5000, S3000	Linux	Syscfg	Smi.ko	2.0.2	GPL, in Intel Carrier Grade Server PDK
S5000, S3000	Microsoft Windows Preinstallation Environment 2005	Syscfg	Smi.sys		
S5000	Linux	Intel OFU (afu)	amifldr_mod.o	V 4.13	Buildable, AMI* proprietary
S5000	Microsoft Windows Preinstallation Environment 2005	Iflash32	Flashud.sys		Binary, Intel
S5000, S3000	Linux	Ethtool	E1000.ko		GPL
S3000	Linux	SW RAID Hypercfg and Spy	Megaide.ko	ich7 – v.5.09	Binary, LSI proprietary
S3000	Microsoft Windows Preinstallation Environment 2005	SW RAID Hypercfg and Spy	Medaide.sys	ichx v4.4.0630.2006	Binary, LSI proprietary
S3000	Microsoft Windows Preinstallation Environment 2005	SW RAID Raidcfg32	lastor.sys	7.5.0.1017	Binary, Matrix proprietary
S5000	Linux	SW RAID Cmdtool2	Megasr.ko	esrt2 07.19.0430.2007	Binary, LSI proprietary

Intel® Server Platform Family	Operating System	Tool	Driver	Version	Location of Source
S5000	Microsoft Windows Preinstallation Environment 2005	SW RAID Cmdtool2	Megasr.sys	esrt2 07.19.0430 .2007	Binary, LSI proprietary

## How To Load Drivers

This section provides example scripts for installing the drivers. You may need to modify the scripts to fit your particular configuration.

In Linux, the general procedure in rc.d or other scripts is:

```
# insmod $driverdir/imb.ko
# maj=`cat /proc/devices | awk '/imb/{print $1}'`
# /bin/mknod /dev/imb c $maj 0
```

In Windows or Windows Preinstallation Environment 2005, to dynamically load a driver, do this (after booting):

```
> copy /y imbdrv.sys %SystemRoot%\System32\Drivers
> win2kinstall %fwupdir%\imbdrv.inf *IMBDRV
```

For Windows Preinstallation Environment 2005, to load default drivers, do the following (during the build):

```
> copy %PDK%\win32\drivers\amt\*. * %PDK%\win32\drivers
> drvinst.exe /q /inf:%PDK%\win32\drivers
/oscd:%W2K3SP1_CD_MOUNT% %NEWPE%
```

## Sample Script Flow

The following table shows a typical flow (from top to bottom) for a provisioning script. The system state is preserved using the persistVariable tool.

**Table 5. Sample Script Flow**

System State	Intel® Server Board S3000 Family		Intel® Server Board S5000 Family	
	Microsoft Windows Preinstallation Environment 2005	Linux	Microsoft Windows Preinstallation Environment 2005	Linux
new	persistVariable update DepState=Sflash			
Sflash	Flash BIOS and Firmware			
	iflash32 eeupdate.exe**	flashupdt eeupdate32**	iflash32, fwpiaupd, frusdr	flashupdt
Sconfig	Configure BIOS and Firmware			



System State	Intel® Server Board S3000 Family		Intel® Server Board S5000 Family	
	Microsoft Windows Preinstallation Environment 2005	Linux	Microsoft Windows Preinstallation Environment 2005	Linux
	syscfg amtcfg	syscfg amtcfg	syscfg ipmiutil	syscfg ipmiutil
Sdisk	Configure Disks			
	hdparm	hdparm	hdparm or sgmode	hdparm or sgmode
Sswraid	Configure SW RAID			
	Hyperwin and Spy Raidcfg32	Hypercfg and Spy	syscfg CmdTool2	syscfg CmdTool2
Shwraid	Configure HW RAID			
	N/A		syscfg CmdTool2	syscfg CmdTool2
Sos	Install Operating System†			
	Ghost* or similar tool	Ghost or similar tool	Ghost or similar tool	Ghost or similar tool
Snet	Configure Network			
	nicSPEED.vbs	ethtool	nicSPEED.vbs	ethtool
Srestore	Restore to factory			
	erase	erase	erase	erase
(finished)	Evaluate Log			

\*\* Available separately from Intel for OEM customers.

† Norton Ghost is not included in the toolkit.

## Installation

The development kit is installed by unzipping the distribution Zip file onto the developer's workstation. After unzipping, you should see the following folders:

### 📁 Toolkit

📁 Linux	Utilities, libraries, and drivers for Linux.
📁 Windows	Utilities, libraries, and drivers for Windows.
📁 Common	Files common to Linux and Windows.
📁 Doc	Documentation and sample scripts.
📁 Demos	Deployment demonstrations for Linux and Windows. This folder also contains documentation on how to create a bootable Windows Preinstallation Environment CD with RAM Disk.

# Runtime Environment

The tools in the Intel Carrier Grade Server PDK are run on the supported platforms once they are on media available to the deployment Operating System. The *bin*, *Python*, *xml*, and *Linux* or *win32* folders are copied to the deployment media. Most of the PDK scripting support tools are implemented in Python for Operating System independence. Most Linux systems have Python available. The Linux installation environment, *Anaconda*, is implemented in Python, so the Intel Carrier Grade Server PDK's scripting support tools integrate well in this deployment environment. Python must be added to, and installed by you, into your Windows Preinstallation Environment Operating System environment in order for the scripting support tools to work.

## Set Deployment Environment Variables (setDeploymentEnv)

A user-customizable Python script called *setDeploymentEnv* is provided to facilitate the setting of the PDK environment variables required in either Windows or Linux. To automatically initialize the required PDK environment variables, *setDeploymentEnv* must be invoked by providing its complete path.

Invoke *setDeploymentEnv* for Linux by using backquotes:

```
`/opt/deploy/bin/setDeploymentEnv`
```

Invoke *setDeploymentEnv* for Windows by calling the provided command file:

```
call %SystemDrive%\deploy\bin\setDeploymentEnv.cmd
```

All of the Intel Carrier Grade Server PDK environment variables shown in the table below are initialized based on the complete path provided when *setDeploymentEnv* is invoked.

## List of Environment Variables

The following table lists the environment variables used by the tools.

**Table 6. Environment Variables**

Variable	Description
DeployPath <sup>†</sup>	The location of the peer directories <i>bin</i> , <i>Python</i> , <i>xml</i> , and <i>Linux</i> or <i>win32</i> . This variable is set automatically by default based on the complete path used to invoke <i>setDeploymentEnv</i> . You may customize this path for your environment, but it must point to the peer directories listed above.
PATH <sup>†</sup>	The search path variable must include the <i>bin</i> and <i>Linux</i> or <i>win32</i> and <i>win32/dlls</i> directories. This variable is appended to based on the complete path used to invoke <i>setDeploymentEnv</i> . You may customize this path for your environment, but it must point to the peer directories listed above.

Variable	Description
LD_LIBRARY_PATH <sup>†</sup>	This must be set to the <i>Linux/libs</i> directory for a Linux system. This variable is appended to based on the complete path used to invoke <code>setDeploymentEnv</code> . You may customize this path for your environment, but it must point to the <i>Linux/libs</i> directory.
PYTHONPATH <sup>†</sup>	The Python search path variable must include the Python directory. This variable is appended to based on the complete path used to invoke <code>setDeploymentEnv</code> . You may customize this path for your environment, but it must point to the Python directory.
DeploymentStore <sup>†</sup>	<p>The path to a persistent store location where the log and hardware inventory files are maintained. You may set this path prior to invoking <code>setDeploymentEnv</code>. When the provided path is not valid, the value of <code>DeploymentStore</code> will be reassigned using the method described below.</p> <p>Reassignment method: First <code>setDeploymentEnv</code> attempts to locate a suitable storage location by searching for the provided path on the other mounted writable devices. Next, if the provided path cannot be found on any other device, a root folder named <code>DeploymentStore</code> is created.</p> <p>For a more information, see “Important Environment Variable Behavior.”</p>
StoreDevice	The drive letter (in Windows) or the mount path (in Linux) of the <code>DeploymentStore</code> . This variable is added to your environment after invoking <code>setDeploymentEnv</code> .
BootDevice	The drive letter (in Windows) or the mount path (in Linux) of the current boot device. This variable is added to your environment after invoking <code>setDeploymentEnv</code> .

<sup>†</sup>These variables must be set correctly before invoking `prepareDeployment`. Use `setDeploymentEnv` to do this for you.

## Prepare Deployment Environment (prepareDeployment)

After the required environment variables are set, the first script normally run is `prepareDeployment` which creates a hardware inventory of the system and loads any drivers that the Intel Carrier Grade Server PDK's tools are dependent on. The tools `queryProperty` and `countProperty` use this hardware inventory.

The toolkit includes two scripts that setup and run `prepareDeployment`. To setup the Python environment, run one of the following commands:

- `rc.local` for Linux environments
- `deploy.cmd` for Windows environments

The following example shows the rc.local script:

```
#!/bin/bash
# /etc/rc.d/rc.local: initiates automatic deployment scripts for Intel platforms
#
# DeploymentStore should be set to a writable and persistent directory location
# setDeploymentEnv finds a SLAX USB device if $DeploymentStore does not exist
# setDeploymentEnv assumes 'sdx1_removable' is in the path
export DeploymentStore=/mnt/sdx1_removable/DeploymentStore
# setDeploymentEnv resets environment variable DeploymentStore in the likely event the
path above is invalid
# setDeploymentEnv sets $DeployPath when setDeploymentEnv is invoked with the full
path
~/opt/deploy/bin/setDeploymentEnv`
ldconfig
prepareDeployment
if [ $? -eq 0 ]
then
    addLogRecord rc.local INFO "Deployment started"
    customDeployment
fi
evaluateDeployment
ret=$?
echo "finished deployment with error level $ret"
addLogRecord rc.local INFO "Deployment finished with error level $ret"
```

The following example shows the deploy.cmd script:

```
rem sets environment to run deployment scripts
rem DeploymentStore should be set to a writable and persistent directory location
rem setDeploymentEnv finds a DOS drive letter for a WinPE USB device if
%DeploymentStore% does not exist
rem setDeploymentEnv assumes a DOS drive letter is first character in the path
set DeploymentStore=Z:\DeploymentStore
call %SystemDrive%\deploy\bin\setDeploymentEnv.cmd
call prepareDeployment
if not errorlevel 1 (
    call addLogRecord deploy.cmd INFO "Deployment started"
    call customDeployment
)
call evaluateDeployment
set ret=%ERRORLEVEL%
echo "finished deployment with error level %ret%"
call addLogRecord deploy.cmd INFO "Deployment finished with error level %ret%"
```

## Customizing a Deployment

The following Python scripts are normally invoked when deploying a system:

- bin/setDeploymentEnv is a customizable Python script that creates the shell commands to set the required environment variables for the deployment scripts
- bin/prepareDeployment initializes the log, does the hardware detect, and dynamically loads drivers. This script is required and should not be modified.
- bin/customDeployment is the customizable script that can be built-up to achieve the deployment objectives.
- bin/evaluateDeployment checks the log and displays the highest severity level found in the log.

## Important Environment Variable Behavior

During development with the Intel Carrier Grade Server PDK tools, you may move from one system configuration to another, each with different numbers and types of USB devices, and different hard drive configurations. You may want to use the same Intel Carrier Grade Server PDK USB memory device or CD-ROM that you have prepared on all of these systems. If using a USB memory device, for instance, to store your Intel Carrier Grade Server PDK results, the path that you provide for DeploymentStore may not be valid on every system, depending on the number of drives and other devices, because the USB drive letter might be different. In this instance, setDeploymentEnv will attempt to locate a suitable storage location and set the value of the DeploymentStore variable.

The setDeploymentEnv.cmd or setDeploymentEnv scripts are used to export the variables to the current shell environment. The exported environment variables are required by the deployment scripts. You should customize setDeploymentEnv for your deployment environment.

Typical output of setDeploymentEnv in Linux:

```
(root)/root# /opt/deploy/bin/setDeploymentEnv
export DeployPath=/opt/deploy
export
PATH=/opt/deploy/bin:/opt/deploy/Linux:/usr/local/sbin:/usr/sbin:/sbin:/usr/local/bin:/usr/bin:/bin:/usr/X11R6/bin:/usr/games:/boot/mybin:/opt/kde/bin:/usr/lib/qt/bin
export LD_LIBRARY_PATH=/opt/deploy/Linux/libs
export PYTHONPATH=/opt/deploy/Python:/opt/deploy/doc
export DeploymentStore=/mnt/sdb1 removable/DeploymentStore
export StoreDevice=/mnt/sdb1 removable
export BootDevice=/mnt/sdb1 removable
(root)/root#
```

Back-quoting the output of the above command exports these variables into the deployment environment. For example:

```
`/opt/deploy/bin/setDeploymentEnv`
```

Typical output of setDeploymentEnv in Windows:

```
X:\i386\SYSTEM32> python /deploy/bin/setDeploymentEnv
set DeployPath=X:\deploy
set
Path=X:\deploy\bin;X:\deploy\win32;X:\deploy\win32\dlls;X:\I386\system32;X:\I386;X:\I386\System32\Wbem;X:\deploy\Python23;C:\bin
set PYTHONPATH=X:\deploy\Python;X:\deploy\doc
set DeploymentStore=C:\DeploymentStore
set StoreDevice=C:
set BootDevice=C:
```

Calling setDeploymentEnv.cmd captures the output of the above command into a temporary command file which it then executes to introduce these variables into the deployment environment. For example:

```
call %SystemDrive%\deploy\bin\setDeploymentEnv.cmd
```

## Tools for Setting Environment Variables in Windows

The Windows shell is not quite as flexible as Linux when setting environment variable to the output of other commands. To help make variable assignment easier, the Intel Carrier

Grade Server PDK provides the varSet.cmd and exportVarSet.cmd scripts.

In a Windows environment, you can use varSet.cmd to create a DOS Shell environment variable that is assigned to the output of other commands. The syntax is:

```
call {path}varSet.cmd VariableName "Command"
```

or, if the command you are evaluating includes double quotation marks, you may pipe the output to the varSet.cmd file and then export the results using exportVarSet.cmd. The syntax in Windows is:

```
Commands | varSet.cmd VariableName  
call exportVarSet.cmd VariableName
```

For example, if this is the syntax in Linux:

```
manuf=`queryProperty SMBIOS Manufacturer - type="2"`  
product=`queryProperty SMBIOS "Product Name" - type="2"`  
version=`queryProperty SMBIOS Version - type="2"`  
serial=`queryProperty SMBIOS "Serial Number" - type="2"`
```

You must use this syntax in Windows (note the use of **varSet** first, then **exportVarSet**):

```
call varSet manuf "queryProperty SMBIOS Manufacturer - type=2"  
  
call queryProperty SMBIOS "Product Name" - type="2" | varSet product  
call exportVarSet.cmd product  
  
call varSet version "queryProperty SMBIOS Version - type=2"  
  
call queryProperty SMBIOS "Serial Number" - type="2" | varSet serial  
call exportVarSet.cmd serial
```

## 2 Utilities

---

This section provides reference information on each utility. Note that some options may require a user response. These options should be avoided in unattended scripts.

### AMTCfg

Configure Intel AMT on Intel Server Platform SR1530AH.

This utility has the following capabilities:

- Enable or disable Intel AMT
- Set or change the Intel AMT password
- Configure the network settings

### Supported Intel® Platforms

- Intel Server Platform SR1530AH

### Supported Environments

- Red Hat Enterprise Linux 4, Update 4 (32-bit x86 Edition only)
- Windows Server 2003 SP1 (32-bit x86 Edition only)
- Windows Preinstallation Environment 2005 (32-bit x86 Edition only)

### Command Line Syntax

```
amtcfg [-deopx -DGHINPS ]
```

where:

-x	Shows extra debug messages
-d	Disables/unprovisions AMT
-e	Enables AMT for remote use
-o <i>old_password</i>	Old Intel AMT password, used with -p (initial password is "admin")
-p <i>password</i>	Sets Intel AMT password (must be a strong password)
-D	Sets DHCP shared mode (default)
-I <i>IP_address</i>	Sets static IP address
-S <i>subnet</i>	Sets Subnet mask (the default is 255.255.255.0)

-G <i>gateway_IP</i>	Sets Gateway IP address
-P <i>dnsIP</i>	Sets Primary DNS address
-N <i>domain</i>	Sets Domain Name
-H <i>hostname</i>	Sets Hostname

**Note:** You must have administrator privileges in Window or root privileges in Linux to run this utility.

## Return Codes

0	Success
1	Error
4	Access denied (user must have administrator or root privileges)

## Script Examples

```
# Display AMT configuration (does not display password
information)
amtcfg
```

```
# Set AMT configuration settings
amtcfg -e -I 192.168.1.2 -G 192.168.1.254 -P 192.168.1.1 -N
intel.com -H srah1 -o admin -p Newpswd~1
```



# CmdTool2

This utility is used to update the firmware for Intel RAID controllers using the SAS software stack.

This tool has the following capabilities:

- Set or display the RAID controller properties
- Set or display the drive properties
- Shutdown the controller
- Configure RAID storage
- Update the firmware
- Display battery information
- Display log information

## Supported Intel® Platforms

- Intel Server Platform SR1530AH
- Intel Server Platform SR2500AL
- Intel® IP Network Server NSW1U
- Intel® Carrier Grade Server TIGW1U
- Intel® IP Network Server NSC2U
- Intel® Carrier Grade Server TIGH2U

## Supported Environments

- Red Hat Enterprise Linux 4, Update 4 (32-bit x86 Edition only)
- Windows Server 2003 SP1 (32-bit x86 Edition only)
- Windows Preinstallation Environment 2005 (32-bit x86 Edition only)

## Dependencies

The server must have a supported Intel RAID controller. The following Intel RAID controllers are supported:

- Intel RAID Controller SRCSAS18E
- Intel RAID Controller SRCSAS144E
- Intel RAID Controller SROMB18E for the Intel Server Board S5000PAL
- Onboard ESB2 and 106x controllers

## Command Line Syntax

Refer to the description of the CmdTool utility in: “Chapter 3 MegaRAID Command Tool,” *MegaRAID SAS Software User’s Guide*, available from LSI Logic at:

[http://www.lsi.com/files/docs/techdocs/storage\\_stand\\_prod/sas/mr\\_sas\\_sw\\_ug.pdf](http://www.lsi.com/files/docs/techdocs/storage_stand_prod/sas/mr_sas_sw_ug.pdf)

## Script Examples

```
#Get Enclosure Device ID
EID="`. ./CmdTool2 -EncInfo -a0 |grep "Device ID"| awk '{print
$4}'`"

#Create RAID 0 Volume using the first 2 disks:
./CmdTool2 -CfgLDAdd -R0[$EID:0,$EID:1] -a0
if [ $? = 1 ];
then
    exit 1
fi

#Initialize the New Volume
./CmdTool2 -LDInit -Start Full -L0 -a0
if [ $? = 1 ];
then
    exit 1
fi# turn off disk write cache
./CmdTool2 -LDSetProp WT -L0 -a0
```

# Disc-Wipe

The dwipe-4.2.exe utility is used to wipe disks clean using multiple passes.

**Caution:** Misuse of this tool may result in unexpected data loss. Please use this tool with caution.

## Supported Platforms

- Intel Server Platform SR1530AH
- Intel Server Platform SR2500AL
- Intel® IP Network Server NSW1U
- Intel® Carrier Grade Server TIGW1U
- Intel® IP Network Server NSC2U
- Intel® Carrier Grade Server TIGH2U

## Supported Environments

- Red Hat Enterprise Linux 4, Update 4 (32-bit x86 Edition only)
- Windows Server 2003 SP1 (32-bit x86 Edition only)
- Windows Preinstallation Environment 2005 (32-bit x86 Edition only)

## Command Line Syntax

Under Windows, type the following to get the syntax information:

```
dwipe.exe -h
```

Under Linux, type the following to get the syntax information:

```
./dwipe -h
```

## Script Examples

```
dwipe -f -s -z /dev/hda
```

# Erase

This tool restores the system to the factory default state. The tool erases disk drives, non-volatile storage, and restores factory default settings.

## Supported Intel® Platforms

- Intel Server Platform SR1530AH
- Intel Server Platform SR2500AL
- Intel® IP Network Server NSW1U
- Intel® Carrier Grade Server TIGW1U
- Intel® IP Network Server NSC2U
- Intel® Carrier Grade Server TIGH2U

## Supported Environments

- Red Hat Enterprise Linux 4, Update 4 (32-bit x86 Edition only)
- Windows Server 2003 SP1 (32-bit x86 Edition only)
- Windows Preinstallation Environment 2005 (32-bit x86 Edition only)

## Command Line Syntax

The Erase utility restores the machine to a default factory state, erasing disk drives, removing non-volatile data, and restoring default configuration values.

**Caution:** Misuse of this tool may result in unexpected data loss. Please use this tool with caution. To help avoid accidental misuse of the tool, erase must be run with a -s switch.

```
erase -s [-u username] [-p password] [-v sda]
```

where:

- |                    |   |
|--------------------|---|
| -s                 | Enable erase function   |
| -u <i>username</i> | Username for RMM2 module (required only if an RMM2 module is present) |
| -p <i>password</i> | Password for RMM2 module (required only if an RMM2 module is present) |
| -v <i>sda</i>      | Volume to skip during the storage cleaning.                           |

## Return Codes

- |   |                    |
|---|--------------------|
| 0 | Success            |
| 1 | Error              |
| 5 | Invalid parameters |

## Script Examples

To erase all components:

```
erase -s
```

To display the command usage:

```
erase
```

# dmidecode

This utility is used by the HW Detect script to report system information from the SMBIOS tables.

## Supported Intel® Platforms

- Intel Server Platform SR1530AH
- Intel Server Platform SR2500AL
- Intel IP Network Server NSW1U
- Intel Carrier Grade Server TIGW1U
- Intel IP Network Server NSC2U
- Intel Carrier Grade Server TIGH2U

## Supported Environments

- Red Hat Enterprise Linux 4, Update 4 (32-bit x86 Edition only)

## Command Line Syntax

Refer to the manual page in Linux for dmidecode (type `man dmidecode`).

# ETHTOOL

This utility configures the networking adapters that are visible to the operating system.

## Supported Intel® Platforms

- Intel Server Platform SR1530AH
- Intel Server Platform SR2500AL
- Intel IP Network Server NSW1U
- Intel Carrier Grade Server TIGW1U
- Intel IP Network Server NSC2U
- Intel Carrier Grade Server TIGH2U

## Supported Environments

- Red Hat Enterprise Linux 4, Update 4 (32-bit x86 Edition only)

## Command Line Syntax

Refer to the manual page in Linux for ethtool.

## Script Examples

```
ethtool -s eth0 speed 1000 duplex full
```

## Additional Documentation

Refer to the manual page in Linux for ethtool.

# FRUSDR

This tool loads FRU and SDR records. (This utility also has other options that are redundant with other tools.)

## Supported Intel® Platforms

- Intel Server Platform SR2500AL
- Intel® IP Network Server NSW1U
- Intel® Carrier Grade Server TIGW1U
- Intel® IP Network Server NSC2U
- Intel® Carrier Grade Server TIGH2U

## Supported Environments

- Windows Preinstallation Environment 2005 (32-bit x86 Edition only)

## Command Line Syntax

```
frusdr [/? | /h] [/p] [/d {smb | fru [options] | sdr}] [/Cfg filename.cfg]
```

options are:

```
frusdr /d fru NVS_type device_ID Dev_ctlr [lun bus address]
```

Where:

<code>/?   /h</code>	Display help
<code>/p</code>	Pause between blocks of data
<code>/d {smb, fru, sdr}</code>	Display SMBIOS (types 1, 2, or 3), FRU, or SDR records
<code>/Cfg <i>filename.cfg</i></code>	Update the FRU or SDR based on the information in the configuration (.cfg) file.
<code>NVS_type</code>	Non-volatile Storage type.
<code>device_ID</code>	Device ID.
<code>Dev_ctlr</code>	Device controller.
<code>lun</code>	Logical Unit Number for the device. The default is the BMC.
<code>bus</code>	Bus Number for the device.
<code>address</code>	Bus address for the device.



## Return Codes and Error Messages

0 Successful completion

1–10 If an error occurs, the FRUSDR utility will print an error message and return a non-zero error code.

Miscellaneous Errors and Warnings	Exit Code of 1
Error, unsuccessful write of FRU File to FRU area	
Error, EEPROM Board area contains an invalid checksum	
Error, EEPROM Chassis area contains an invalid checksum	
Error, environment variable not found	
Error, exceeded maximum allowed BCD string length	
Error, invalid data found in record data portion	
Error, failed to retrieve header information from FRU file	
Error, failed to send message	
Error, insufficient memory	
Error, invalid Binary Coded Decimal (BCD) number was found	
Error, invalid field	
Error, invalid header	
Error, an invalid field length of one was found	
Error, invalid number of bytes used to indicate hex number	
Error, invalid string entered. You must enter a valid ASCII string	
Error, address values must be even	
Error, odd BCD length entered, so a zero was pre-appended	
Error, only 16 FRU addresses are allowed per FRU file	
Error, EEPROM Product area contains an invalid checksum	
Error, processing FRU file	
Error, reading string length	
Error, reading record from file	
Error, reading SDR record from file	
Error, failed to read FRU string	
Error, security of program has been breached	
Error, type not found	
Error, this utility does not support saving the field in the requested type	
Error, all addresses must be unique	
Error, wrong number of bytes read	
Error, in verifying the written string	
Command Line Errors	Exit Code of 2
Error, expected FRU file name	
Error, too many command line arguments	
Error, invalid command line parameter(s)	
Error, invalid DEV Ctrl after NVS Type on the command line	
Error, invalid Device ID found after NVS Type on command line	
Error, invalid file name extension entered	
Error, invalid LUN address found after NVS Type on command line	
Error, only one device address is allowed when using NVS Type of Device ID	
Error, missing command line parameter(s)	
Error, no command line arguments were given	
Error, no filename extension entered	

Error, too many device addresses, only 16 device addresses are allowed	
<b>FRU &amp; SDR Programming Errors</b>	<b>Exit Code of 3</b>
Error, selected area not present	
Error, board FRU data is larger than the board FRU area	
Error, chassis FRU data is larger than the chassis FRU area	
Error, Automatic-iFICS is not available	
Error, Internal Use area size overflow	
Error, Multi-Record area size overflow	
Error, product FRU data is larger than the product FRU area	
Error, failed to clear the SDR Repository	
<b>Configuration File Parsing Errors</b>	<b>Exit Code of 4</b>
Error, argument follows a command in the configuration file when it should not	
Error, argument is too long that follows command in the configuration file	
Error, exceeded maximum argument length	
Error, incorrectly formatted input '@' string	
Error, invalid argument following FRU area in the configuration file	
Error, invalid argument following FRU name in the configuration file	
Error, invalid argument following SDR name in the configuration file	
Error, invalid argument found in configuration file	
Error, invalid baseboard ID string length specified	
Error, invalid Board Area in file	
Error, invalid bus address	
Error, invalid command found in the configuration file on line ##	
Error, invalid comment in configuration file on line ##	
Error, invalid configuration file string found	
Error, invalid Chassis Area in file	
Error, invalid chassis type found, must be a hex number from 01 to 18 hex	
Error, invalid device address found	
Error, invalid field input type specified in the configuration file	
Error, invalid FRU address found in configuration file	
Error, invalid FRU address format in configuration file	
Error, invalid FRU area found in configuration file	
Error, invalid FRU title length was found	
Error, invalid Header Area in file	
Error, invalid Internal Use Area in file	
Error, invalid line number found	
Error, invalid Multi-Record Area in file	
Error, invalid device address, must be an even number	
Error, invalid number of arguments following the FRUADDRESS command	
Error, invalid number of arguments following the FOUND command	
Error, invalid number of arguments following the PROBE command	
Error, invalid PROCESSOR number, valid numbers are 1 through 8	
Error, invalid Product Area in file	
Error, invalid SLOT2CPU number, valid numbers are 0 through 7	
Error, matching quote missing after command in the configuration file	
Error, the FRU file area must come before the FRU field	
Error, a FRU file name was found, but no FRU areas were specified	
Error, no valid argument found after command in the configuration file	
Error, no conditional statement found after IFSET	

Error, invalid FRU field was found	
Error, an IFSET statement must precede the ELSE statement	
Error, an IFSET statement must precede the ENDIF statement	
Error, a MENU statement must precede the MENUPROMPT statement	
Error, the FRU file name must come before the FRU address	
Error, the FRU file name must come before the FRU area	
Error, a PROMPT command must precede the YES and NO commands	
Error, no previous corresponding PROBE for the FOUND command	
Error, non-matching IfSet and Endif statements	
Error, non-matching quotes found on configuration line	
Error, parsing configuration information	
Error, problem in parsing configuration file	
Error, parsing configuration line	
Error, problem in reading configuration file	
Error, too many arguments found after command in the configuration file	
Error, only up to nine MENU commands may precede the MENUPROMPT command	
<b>Protocol / Communication Errors</b>	<b>Exit Code of 5</b>
Error, busy bit timeout	
Error, DATA READY timeout.	
Error, could not detect BMC through ISA driver	
Error, EEPROM not initialized	
Error, failed to get Read End from SMIC	
Error, failed to get Read Start from SMIC	
Error, failed to get Ready from SMIC	
Error, failed to clear the SDR repository	
Error, failed parsing message	
Error, failed receiving message	
Error, failed sending message	
Error, I2C bus protocol problem	
Error, reading EEPROM string	
Error, reading internal thresholds	
Error, failed to receive BMC response	
Error, failed to receive BMC response, bad completion code	
Error, unknown failure while trying to receive BMC response	
Error, failed to send BMC command	
Error, in writing string to EEPROM	
Error, in verifying write to EEPROM	
Error, transmit buffer full	
<b>SM BIOS Problems</b>	<b>Exit Code of 6</b>
Error, INT15 (DA92h) Call not supported for CPU Config Info	
Error, "Get SM BIOS Information" request failed	
Error, "Get SM BIOS Structure" request failed	
Error, invalid SM BIOS type requested: ##	
Error, "Set SM BIOS Structure" request failed	
Error, writing to SM BIOS space	
<b>Verify/Compare Errors</b>	<b>Exit Code of 7</b>
Error, verification of one or more FRU Area(s) failed	
Error, board areas defined in the file and in memory are different	

Error, board area checksums defined in the file and in memory are different	
Error, board area lengths defined in the file and in memory are different	
Error, board area offsets defined in the file and in memory are different	
Error, chassis areas defined in the file and in memory are different	
Error, chassis area checksums defined in the file and in memory are different	
Error, chassis area lengths defined in the file and in memory are different	
Error, chassis area offsets defined in the file and in memory are different	
Error, common header checksums defined in the file and in memory are different	
Error, file and EEPROM header areas must be identical, but are different	
Error, header length does not match with the sum lengths of each area	
Error, header format versions defined in the file and in memory are different	
Error, internal use area lengths defined in the file and in memory are different	
Error, internal use area offsets defined in the file and in memory are different	
Error, MultiRecord Area ## defined in the file and in memory are different	
Error, MultiRecord area offsets defined in the file and in memory are different	
Error, FRU EEPROM MultiRecord Area ## checksum is invalid	
Error, MultiRecord Area ## in FRU File (or Memory) is invalid	
Error, product areas defined in the file and in memory are different	
Error, product area checksums defined in the file and in memory are different	
Error, product area lengths defined in the file and in memory are different	
Error, product area offsets defined in the file and in memory are different	
Error, invalid baseboard ID string specified	
Error, SDR file does not match with what was programmed into memory	
Error, SDR file unsuccessfully written	
Error, FRU EEPROM Common Header Area checksum is invalid	
Error, FRU EEPROM Chassis Area checksum is invalid.	
Error, FRU EEPROM Board Area checksum is invalid.	
Error, FRU EEPROM Product Area checksum is invalid.	
Error, There are more MULTIREC Records in the FRU Memory than in the FRU File	
Error, There are more MULTIREC Records in the FRU File than in the FRU Memory	
FRU and SDR File Parsing Errors	Exit Code of 8
Error, padded Board Area in FRU area on this server must be all zeros	
Error, padded Chassis Area in the FRU area on this server must be all zeros	
Error, expected valid hexadecimal data after _DEV_ADDRESS	
Error, expected valid hexadecimal data after _DEV_BUS	
Error, expected valid hexadecimal data after _DATA_LEN	
Error, expected valid hexadecimal data after _REC_LEN	
Error, expected valid hexadecimal data after _SDR_TYPE	
Error, expected valid hexadecimal data after _START_ADDR	
Error, expected static data after _DEV_ADDRESS	
Error, expected static data after _DATA_LEN	
Error, expected static data after _DEV_BUS	
Error, expected static data after _DEV_CNTR	
Error, expected static data after _DEV_ID	
Error, expected static data after _NVS_LUN	
Error, expected static data after _START_ADDR	
Error, _DATA_LEN should follow _START_ADDR	
Error, expected _DATA_LEN to follow after _START_ADDR	
Error, _DEV_ADDRESS should follow _DEV_BUS	
Error, _DEV_BUS should follow NVS TYPE	

Error, expected _FRU or _SDR tag	
Error, _IPMI_VERSION should follow _LF_FMT_VER	
Error, expected a left parenthesis '('	
Error, _LF_FMT_VER should follow _LF_VERSION	
Error, _LF_NAME should be the first tag in the file	
Error, _LF_VERSION should follow _LF_NAME	
Error, _NVS_TYPE should follow _DATA_LEN	
Error, _REC_LEN should follow _SDR_TAG	
Error, _START_ADDR should be first entry in SEL header info	
Error, _START_ADDR should be first entry in FRU header info.	
Error, expected ASCII text after _FRU_TITLE	
Error, expected ASCII text after _NVS_TYPE	
Error, expected ASCII text after _LF_FMT_VER	
Error, expected ASCII text after _LF_NAME	
Error, invalid or unsupported _NVS_TYPE	
Error, expected ASCII text after _IPMI_VERSION	
Error, expected ASCII text after _LF_VERSION	
Error, FRU Board Area may only be defined once per FRU file	
Error, FRU Chassis Area may only be defined once per FRU file	
Error, FRU Header Area may only be defined once per FRU file	
Error, FRU Internal Use Area may only be defined once per FRU file	
Error, FRU Product Area may only be defined once per FRU file	
Error, invalid Board Area length in the FRU file	
Error, invalid Board Area length in the FRU area on this server	
Error, invalid common header checksum	
Error, invalid Chassis Area length in the FRU file	
Error, invalid Chassis Area length in the FRU area on this server	
Error, invalid environment variable length	
Error, invalid field length found in the FRU file	
Error, invalid FRU file header was found	
Error, invalid Multi-Record Area header block was found	
Error, invalid Product Area length in the FRU area on this server	
Error, invalid Sensor Data Record found, number ##	
Error, invalid Sensor Data Record length found	
Error, no arguments permitted after the Manufacturing Date & Time field	
Error, this program only supports file format version ##	
Error, this program only supports SDR format version ##	
Error, this program only supports SDR format version ## or higher	
Error, parsing SDR file	
Error, padded Product Area in the FRU area on this server must be all zeros	
Error, expected ASCII text after _SDR_TAG	
Error, parsing SDR or FRU file	
Error, unsupported field type found	
Error, invalid Sensor Table Record found, number ##	
Error, invalid Sensor Table Record length found	
File Processing Errors	Exit Code of 9
Error, failed to open CFG.TMP file	
Error, failed to open SMB.TMP file	
Error, failed to open FRU.TMP file	
Error, failed to open SDR.TMP file	
Error, failed to open the default master input file: MASTER.CFG	

Error, failed to open the default master input file: MASTER.FRU	
Error, failed to open the default master input file: MASTER.SDR	
Error, failed to open the specified input file	
Error, failed to open the specified configuration file	
Error, failed to open the specified FRU file	
Error, failed to open the specified SDR file	
Error, an invalid character was found on line	
Error, an invalid character was found on line ##	
Error, invalid file size. FRU, SDR and CFG files must be less than 256 Kbytes	
Error, line ## is too long. Lines lengths must be 255 characters or less	
Error, failed when writing to temporary file	
<b>General Error</b>	<b>Exit Code of 10</b>
(none)	

Note:

An error exit code of 10 will be returned in some cases. Typically this will happen when more than one error has occurred.

## Script Examples

```
FRUSDR /cfg \path\MASTER.CFG
```

# FWPIAUPD

This utility is used to update the boot, Op Code, and PIA blocks for the BMC, Intel LCP, and HSC firmware.

## Supported Intel® Platforms

- Intel Server Platform SR2500AL
- Intel® IP Network Server NSW1U
- Intel® Carrier Grade Server TIGW1U
- Intel® IP Network Server NSC2U
- Intel® Carrier Grade Server TIGH2U

## Supported Environments

- Windows Preinstallation Environment 2005 (32-bit x86 Edition only)

## Command Line Syntax

### BMC Updates

**FWPIAUPD** [-?|-h] [-ni] [-u|-v] [-b] [-o] [-pia] [-p|-s] [-nodownrev] [-i [-address=xx]] [-errlog=file] [-setdef] [hexfile]

FWPIAUPD

The name of the utility.

*hexfile*

The name of the hex file for the operation. The path can be specified with the filename, e.g. 'c:\mydir\updatfil.hex'. There is no default filename or extension.

/? Or /help

Displays the command line help.

/ni

Forces non-interactive mode. If the specified operation requires user input or if the operation is not specified completely the utility will exit with an error.

/nodownrev

Prevents the utility from uploading a lower revision of firmware than present in the system.

/upload

Uploads and verifies the firmware.

/verify

Verifies the firmware.

/boot

Applies the specified operation to the boot area. The operational and PIA (if present in

	the file) areas are also updated/verified.
<code>/op</code>	Applies the specified operation to the operational area. The PIA area (if present in the file) is also updated/verified.
<code>/pia</code>	Applies the specified operation to the PIA area only. This will only effect the BMC at address 20.
<code>/primary</code>	Selects the primary controller to update/verify with the specified file.
<code>/secondary</code>	Selects the secondary controller to update/verify with the specified file.
<code>/inventory</code>	Shows the firmware versions from the specified hex file. If no hex file is specified the <code>-address</code> option, or the default 20 is used. Shows the BMC, Firmware, and PIA versions from a file; and BMC and Firmware versions from the controller.
<code>/address=XX</code>	Specifies the optional address to use for the inventory command. The address is specified in Hexadecimal. The default address is 20.
<code>/errlog=file</code>	Set a file that the error stack should be written to in the event an error occurs. When there are no errors "successful termination" is written to the file. The command will overwrite an existing file. This file will be useful for support to troubleshoot any problems that may occur.
<code>/setdef</code>	Restores the BMC Firmware non volatile settings and the currently configured FW settings will be overwritten with default values. The switch should be used judiciously.

## HSC updates

**FWPIAUPD** `[-?|-h] [-ni] [-u|-v] [-b] [-o] [-dirhsc] [-p|-s] [-nodownrev] [-i [-address=XX]] [-errlog=file] [-setdef] [hexfile]`

FWPIAUPD                      The name of the utility.



<code>&lt;hexfile&gt;</code>	The name of the HSC firmware hex file for the operation. The path can be specified with the filename, e.g. 'c:\mydir\updatfil.hex'. There is no default filename or extension.
<code>/? Or /help</code>	Displays the command line help.
<code>/ni</code>	Forces non-interactive mode. If the specified operation requires user input or if the operation is not specified completely the utility will exit with an error.
<code>/nodownrev</code>	Prevents the utility from uploading a lower revision of firmware than present in the system.
<code>/upload</code>	Uploads and verifies the firmware.
<code>/verify</code>	Verifies the firmware.
<code>/boot</code>	Applies the specified operation to the boot area. For Vitesse HSC only
<code>/op</code>	Applies the specified operation to the operational area. For Qlogic HSC only.
<code>/primary</code>	Selects the primary controller to update/verify with the specified file.
<code>/secondary</code>	Selects the secondary controller to update/verify with the specified file.
<code>/inventory</code>	Shows the firmware versions from the specified hex file. If no hex file is specified the <code>-address</code> option, or the default C0 is used. Shows the HSC Firmware version from a file and from the controller.
<code>/address=XX</code>	Specifies the optional address to use for the inventory command. The address is specified in Hexadecimal. The default address is C0.

<code>/dirhsc</code>	<p> Informs the utility to undertake direct HSC update operation. </p>
<code>/errlog=file</code>	<p> Set a file that the error stack should be written to in the event an error occurs. When there are no errors "successful termination" is written to the file. The command will overwrite an existing file. This file will be useful for support to troubleshoot any problems that may occur. </p>

## Intel LCP Updates

**FWPIAUPD** [-?|-h] [-ni] [-u|-v] [-b] [-o] [-p|-s] [-i[-address=xx]] [-errlog=file] [*hexfile*]

<code>FWPIAUPD</code>	<p> The name of the utility. </p>
<i>hexfile</i>	<p> The name of the hex file for the operation. The path can be specified with the filename, e.g. 'c:\mydir\updatfil.hex'. There is no default filename or extension. </p>
<code>/? Or /help</code>	<p> Displays the command line help. </p>
<code>/ni</code>	<p> Forces non-interactive mode. If the specified operation requires user input or if the operation is not specified completely the utility will exit with an error. </p>
<code>/upload</code>	<p> Uploads and verifies the firmware. </p>
<code>/verify</code>	<p> Verifies the firmware. </p>
<code>/op</code>	<p> Applies the specified operation to the operational area. </p>
<code>/inventory</code>	<p> Shows the firmware versions from the specified hex file. If no hex file is specified the -address option is used. Shows the LCP Firmware version from a file and from the controller. </p>

<code>/address=xx</code>	Specifies the optional address to use for the inventory command. The address is specified in Hexadecimal.
<code>/errlog=file</code>	Set a file that the error stack should be written to in the event an error occurs. When there are no errors "successful termination" is written to the file. The command will overwrite an existing file. This file will be useful for support to troubleshoot any problems that may occur.

## Error Messages and Return Codes

Error Code	Error Message
0	Successful termination.
1	Invalid Parameters.
2	Hex file not found.
3	Invalid hex file.
4	Microcontroller not found in system.
5	Update failure.
6	Verify failure.
7	Invalid hex file for this platform.
8	Could not enter FW transfer mode.
9	Could not exit FW transfer mode.
10	Out of memory.
11	Unknown Boot Device Info Block.
12	Unknown Op Device Info Block.
13	Unknown PIA Header.
14	User aborted the operation.
15	No areas to update.

16	No areas to verify.
17	Invalid Argument.
18	User aborted setting the action.
19	User aborted setting the area(s).
20	User aborted setting a filename.
21	Invalid filename.
22	User aborted upload.
23	Verify byte mismatch.
24	Error reading from file.
25	Invalid system flash.
26	Unsupported extended segment address record.
27	Unsupported start segment address record.
28	Unsupported start linear address record.
29	Unknown record type.
30	Data records before extended linear address record.
31	Multiple extended linear address records for the same address.
32	Extended linear address record out of range.
33	Set segment error.
34	Read error.
35	Read error.
36	Write error.
37	Get mode error.
38	Get version error.
39	Range checksum error.
40	Unexpected command returned from IPMI read.
41	Completion code error.
42	Unable to get IPMI message.
43	Set segment error.
44	Set segment failure.

45	Unable to downrev FW.
46	Invalid chip erase command.
47	Write size change failure.
48	Data buffer get failure.
49	PCI BIOS support unavailable.
50	SMBus timeout.
51	Failed to get SMBus ownership.
52	SMBus bus error.
53	SMBus device error.
54	Could not detect BMC.
55	Invalid boot block hex file.
56	Invalid firmware hex file.
57	Product mismatch in boot block and firmware hex files.
58	Boot block and firmware hex files are for unknown product.
59	Version mismatch in boot block and firmware hex files.
60	CRC mismatch.
61	CRC Failure.
62	Error writing data to mini BMC.
63	Cannot upgrade from firmware version less than 2.10
64	SMBUS send failure.
65	SMBUS read failure.
66	SMBUS Command or sequence mismatch in request and response.
67	HSC busy.
68	SMBus Interface detection failure.
69	HSC has been Reset.
70	HSC Reset Failed.
72	Cannot check for platform mismatch.
73	Failed to restore non volatile configuration.
74	Boot Block update not supported.

- 75 Platform check failed.
- 76 Failed to synchronize SMBus.

## Script Examples

```
fwpiaupd /dirhsc /u /op /p sample.hex (direct HSC)
```

```
fwpiaupd /u /op /p sample.hex (indirect HSC)
```

```
fwpiaupd /ni /u /b /p sample.hex
```

The following example will update the Intel LCP firmware.

```
fwpiaupd sample.hex
```

To update the string hex file ,following command needs to be used.

```
fwpiaupd /nopc /bysegment /address=22 stringFile.hex
```

To do a sample firmware update for the Intel Server Board S5000PAL:

```
fwpiaupd -ni -p -u -o -pia -errlog=\frusdr.log \fw-s5\HB_BMC16.HEX
```

# hdparm

This utility is used to display and set the hard disk drive parameters for ATA disks.

## Supported Intel® Platforms

- Intel Server Platform SR1530AH
- Intel Server Platform SR2500AL
- Intel® IP Network Server NSW1U
- Intel® Carrier Grade Server TIGW1U
- Intel® IP Network Server NSC2U
- Intel® Carrier Grade Server TIGH2U

## Supported Environments

- Red Hat Enterprise Linux 4, Update 4 (32-bit x86 Edition only)
- Windows Server 2003 SP1 (32-bit x86 Edition only)
- Windows Preinstallation Environment 2005 (32-bit x86 Edition only)

## Command Line Syntax

```
hdparm [options] [device] ..
```

Options:

- |          |  |
|----------|--|
| -a       | get/set fs read ahead                                |
| -A       | set drive read-lookahead flag (0/1)                  |
| -b       | get/set bus state (0 == off, 1 == on, 2 == tristate) |
| -B       | set Advanced Power Management setting (1-255)        |
| -c       | get/set IDE 32-bit IO setting                        |
| -C       | check IDE power mode status                          |
| -d       | get/set using_dma flag                               |
| --direct | use O_DIRECT to bypass page cache for timings        |
| -D       | enable/disable drive defect management               |
| -E       | set cd-rom drive speed                               |
| -f       | flush buffer cache for device on exit                |
| -g       | display drive geometry                               |
| -h       | display terse usage information                      |

-i	display drive identification
-l	detailed/current information directly from drive
--lstdin	read identify data from stdin as ASCII hex
--lstdout	write identify data to stdout as ASCII hex
-k	get/set keep_settings_over_reset flag (0/1)
-K	set drive keep_features_over_reset flag (0/1)
-L	set drive doorlock (0/1) (removable harddisks only)
-M	get/set acoustic management (0-254, 128: quiet, 254: fast) (EXPERIMENTAL)
-m	get/set multiple sector count
-n	get/set ignore-write-errors flag (0/1)
-p	set PIO mode on IDE interface chipset (0,1,2,3,4,...)
-P	set drive prefetch count
-q	change next setting quietly
-Q	get/set DMA tagged-queuing depth (if supported)
-r	get/set device readonly flag (DANGEROUS to set)
-R	register an IDE interface (DANGEROUS)
-S	set standby (spindown) timeout
-t	perform device read timings
-T	perform cache read timings
-u	get/set unmaskirq flag (0/1)
-U	un-register an IDE interface (DANGEROUS)
-v	defaults; same as -mcudkrag for IDE drives
-V	display program version and exit immediately
-w	perform device reset (DANGEROUS)
-W	set drive write-caching flag (0/1) (DANGEROUS)
-x	tristate device for hotswap (0/1) (DANGEROUS)
-X	set IDE xfer mode (DANGEROUS)
-y	put IDE drive in standby mode
-Y	put IDE drive to sleep



- Z                      disable Seagate auto-powersaving mode
- z                      re-read partition table
- security-help       display help for ATA security commands

**Caution:** some hdparm options are marked DANGEROUS or EXPERIMENTAL by the developers. If used incorrectly, these options may cause unexpected data loss.

## Return Codes

- 0        Success
- 1        Error
- 1       Failure

## Script Examples

```
# Display disk information
hdparm -I /dev/hda
# Set Write Cache off
hdparm -W0 /dev/hda
```

## Additional Documentation

Refer to the hdparm manual page in Linux (command “man hdparm”), or to the documentation included with the utility for the Windows version.

# HyperCfg and Spy

HyperCfg and Spy are used to configure software RAID on Intel Server Platform SR1530AH running Red Hat Linux. See hyperwin for the Linux version of this tool.

## Supported Intel® Platforms

- Intel Server Platform SR1530AH

## Supported Environments

- Red Hat Enterprise Linux 4, Update 4 (32-bit x86 Edition only)

## Command Line Syntax

hypercfg /[Options]

Options :

```
/ADD /A<m>:d1,d2...dn [/G[n]:<logdrv size> [/G[n]:<logdrv size>] ...]- Configure
- arrays for Raid 0,1 & 10. Allows Auto configuration too.
- Where 'ADD' adds the new Lds to the existing Lds.
- Where 'm' is Default RAID Mode for configuring.
- When 'm' = S[n], Arrays Configured for Raid 0 mode,'n'- Stripe size.
- Allowed Stripe Sizes:8k,16k,32k,64k,128k.
- When 'm' = M, Arrays configured for Raid 1 mode.
- When 'm' = R[n], Arrays configured for Raid 10 mode.'n'- Stripe size.
- Allowed Stripe Sizes:8k,16k,32k,64k,128k.
- D1, D2,...Dn are the drive ids that participate in the array
- /G[n]:<Logdrv size> /G[n]:<Logdrv size>- n is optional and type of Raid
- mode for multiple/mixed mode Raid creation restricted to R0 and R1.
- n = s for R0 and n = m for R1 and default Raid mode will be overridden.
- Default Raid mode will be taken if n is not specified.
- <Logdrv size> is logical drive in MB to be created.
- Note the value is in DEC
- If user does not provide /G switch, Lds will be created as Auto Config.
/C<n> - Selects card where the drive is present.
- Where 'n' is Card index starting from 0,1,2..and so on.
/D<cd> - Selects drive for R, I, L, E, P Options.
- Where 'c' is Channel ID taking values 0 or 1
- Where 'd' is Device ID taking values 0 or 1.
/L [/C<n>] [/D<cd>] [/b[filename]]- Displays the configuration sector of specified
drive.
- using only /L displays the config. sector from first available
- Drive.
- Use /bfilename to dump the IRCD as bin file
- filename is not specified, HYPERCFG.BIN is default
```

```

/E<o> [/C<n>] [/D<cd>] - Erases & Displays the config. sector of specified drive.
    - Where 'o' specifies option for erasing.
    - when 'o' = c, Erases config sector only.
/F<fn> - Redirects the output to a file 'fn'.
    - Default filename is 'hypercfg.cfg'.
/Y[x][fn] - Logs to the given filename fn
    - when 'x' = f, logging is done only for errors
    - when 'x' = c, continuous logging is done
    - Default is 'x' = f and filename is HYPERCFG.LOG
/S - Executes in Silent mode.
/V - Displays the RAID bios version.
/X - Pause execution on encountering an error conditon.
/@ /b<bfm> - Configures RAID from specified file .
    - /@ /b<bfm> - config from bin file bfm Default File:hypercfg.bin
/INIT /l<n1> [/l<n2>] ... - Initializes the specified LDs.
    - Where n1, n2,... specifies LD numbers.
/RBLD /d<nn> - Rebuilds the specified PD.
    - Where <nn> specifies the PD Id.
/CON /l<n1> [/l<n2>] ... - Perform CheckConsistency on the specified LDs.
    - Where n1, n2,... specifies LD numbers.
/PDSTATE /d<nn> /<OFF/ON> - Changes the state of the specified PD.
    - Where <nn> specifies the PD Id.
    - Where <OFF/ON> specifies the new PD state.
/ADP /C<n> - Provides the specified Controller Info.
    - Where <n> specifies the Adapter number.
/PDINFO /C<n> - Provides the PD info of the specified Controller.
    - Where <n> specifies the Adapter number.
/FREEARR /A<m>:d1,d2,dn.. - Provides the Free Array Info.
    - Where <m> specifies the Raid mode user needs to create.
    - Where d1, d2, dn... are the drive list.
/SETADP /C<n> [/r<val> ] [/n<val>] [/bios<-on/off>]
    [/s<-on/off>] [/f<-on/off>] [/cc<val>] [/ars<-on/off>]
    - [/boot<val>] [/arb<-on/off>]
    - Allows user to set values to adapter enquiry information.
    - /C<n> is the controller index.
    - /r<val> set the rebuild rate of specified controllers.
    - <val> should be 0-100
    - /bios<-on/off> to enable/disable bios.
    - /s<-on/off> is a bios stop on error setting.
    - /n<val> sets the action if a new device is found.
    - <val> 0: Auto-config as Spare or R0 (depend on RAID level)
    - 1: Force into ^M.
    - /f<-on/off> enable/disable fast initialization
    - /cc<val> set action to be taken during check consistency operation.
    - <val> 0: Fix/Report, 1:Report only.
    - /ars<-on/off> enable/disable auto resume feature.

```

- /boot<val> set the bootable logical drive. <val> = LD number.
- /arb<-on/off> enable/disable auto rebuild feature.

/SETPD /C<n> [/wc<-on/off> ] [/ra<-on/off>]

- Allows user to set values to Physical drive enquiry information.
- /C<n> is the controller index.
- /wc<-on/off> enable/disable disk write cache.
- /ra<-on/off> enable/disable disk read ahead.

/DEL /l<n1> [/l<n2>] ... - Perform Ld deletion on the specified LDs.

- Where n1, n2,... specifies LD numbers.

/SETFACT /C<n> - Set the factory default.

- /C<n> is the controller index.

## Return Codes

Exit code returned can be modified using qualifer /Q.

When /Q is specified, config. related commands return ARRAY COUNT on SUCCESS. In all other cases ERROR LEVEL is returned

All the other options except /S, /F, /B, /@ & /W will have default arg as zero.

Error Codes :

6h - Array not formed due to inadequate drives

- This error occurs if adequate drives are not present
- for raid to be configured.

0Ah - Hard disk(s) not found

- This error occurs when required hard disks are not found

0Bh - Raid controller timed out on read/write

Possible reasons are damaged HDD, invalid drive id specified

0Dh - Error Opening Array Information file

- Configuration file specified could not be opened

0Eh - Array information file format incorrect

Possible reasons, missing fields/wrong information in file

10h - Error Opening Log file

- Log file name/path may be wrong/invalid

12h - Array not formed as required drives not found

- This error occurs if required drives are not found
  - 13h - Array not formed, excess drives found
    - This error occurs when more than 2 drives present in /! command
  - 16h - Invalid configuration present
    - This error occurs when no raid config. is present and
    - /L command is issued
  - 17h - Configuration mismatch in drives
    - This error occurs when the config. info in all drives do not match. Occurs with /T command
  - 18h - Error reading file
  - 19h - Error writing file
  - 21h - Physical Drive Rebuild Failed
  - 22h - Check Consistency Failed
  - 23h - No FreeArray Info Available
  - 24h - Physical Drive Rebuild Aborted
  - 25h - Check Consistency Aborted
  - 26h - Ld Initialization Aborted
  - 27h - Failed to connect spy
  - 28h - Logical drive deletion failed
  - 1Ah - Warning!! Truncating file
    - This error occurs when the file dumped to a sector is greater than 512 bytes
  - 1Bh - Raid BIOS not found
    - Occurs with /V command if raid bios is not found
  - 1Ch - Feature not available in IDE drive
    - Error occurs if the /P or /M command is used on drives
    - that do not support this feature, or invalid params. are specified
- Other error codes are reserved

## Script Examples

```
#Returns information about the RAID adapter  
./hypercfg /ADP /C0
```

```
#Clears previous RAID volumes and configuration information and  
creates RAID1 on ports #0 and #1 without asking for user input.  
./hypercfg /AM:00,10 /S
```

```
#Initialize Volume 0  
./hypercfg /INIT /10
```

```
#Set Disk Cache options for Port#0 and Port #1  
./hypercfg /SETPD /C0 /wc-on /ra-on  
./hypercfg /SETPD /C1 /wc-on /ra-on
```

# Hyperwin and Spy

Hyperwin and Spy are used to configure software RAID on Intel Server Platform SR1530AH running Windows. See hypercfg for the Linux version of this tool.

## Supported Intel® Platforms

- Intel Server Platform SR1530AH

## Supported Environments

- Windows Server 2003 SP1 (32-bit x86 Edition only)
- Windows Preinstallation Environment 2005 (32-bit x86 Edition only)

## Command Line Syntax

hyperwin /[Options]

Options :

```
/ADD /A<m>:d1,d2...dn [/G[n]:<logdrv size> [/G[n]:<logdrv size>] ...]- Configure
- arrays for Raid 0,1 & 10. Allows Auto configuration too.
- Where 'ADD' adds the new Lds to the existing Lds.
- Where 'm' is Default RAID Mode for configuring.
- When 'm' = S[n], Arrays Configured for Raid 0 mode,'n'- Stripe
size.
- Allowed Stripe Sizes:32k,64k.
- When 'm' = M, Arrays configured for Raid 1 mode.
- When 'm' = R[n], Arrays configured for Raid 10 mode.'n'- Stripe
size.
- Allowed Stripe Sizes:32k,64k.
- D1, D2,..Dn are the drive ids that participate in the array
- /G[n]:<Logdrv size> /G[n]:<Logdrv size>- n is optional and type of
Raid
- mode for multiple/mixed mode Raid creation restricted to R0 and
R1.
- n = s for R0 and n = m for R1 and default Raid mode will be
overridden.
- Default Raid mode will be taken if n is not specified.
- <Logdrv size> is logical drive in MB to be created.
- Note the value is in DEC
- If user does not provide /G switch, Lds will be created as Auto
Config.
/C<n> - Selects card where the drive is present.
- Where 'n' is Card index starting from 0,1,2..and so on.
/D<cd> - Selects drive for R, I, L, E, P Options.
- Where 'c' is Channel ID taking values 0 or 1
- Where 'd' is Device ID taking values 0 or 1.
```

```

/L [/C<n>] [/D<cd>] [/b[filename]] - Displays the configuration sector of
specified drive.
    - using only /L displays the config. sector from first available
    - Drive.
    - Use /bfilename to dump the IRCD as bin file
    - filename is not specified, HYPERCFG.BIN is default
/E<o> [/C<n>] [/D<cd>] - Erases & Displays the config. sector of specified drive.
    - Where 'o' specifies option for erasing.
    - when 'o' = c, Erases config sector only.
/F<fn> - Redirects the output to a file 'fn'.
    - Default filename is 'hypercfg.cfg'.
/Y[x][fn] - Logs to the given filename fn
    - when 'x' = f, logging is done only for errors
    - when 'x' = c, continuous logging is done
    - Default is 'x' = f and filename is HYPERCFG.LOG
/S - Executes in Silent mode.
/V - Displays the RAID bios version.
/X - Pause execution on encountering an error condition.
/@ /b<bfn> - Configures RAID from specified file .
    - /@ /b<bfn> - config from bin file bfn Default File:hypercfg.bin
/INIT /l<n1> [/l<n2>] ... - Initializes the specified LDs.
    - Where n1, n2,... specifies LD numbers.
/RBLD /d<nn> - Rebuilds the specified PD.
    - Where <nn> specifies the PD Id.
/CON /l<n1> [/l<n2>] ... - Perform CheckConsistency on the specified LDs.
    - Where n1, n2,... specifies LD numbers.
/PDSTATE /d<nn> /<OFF/ON> - Changes the state of the specified PD.
    - Where <nn> specifies the PD Id.
    - Where <OFF/ON> specifies the new PD state.
/HSPPD /d<nn> /<OFF/ON> - Set/Reset the state of the specified PD for
    Hot Spare operation.
    - Where <nn> specifies the PD Id.
    - Where <OFF/ON> specifies the PD state changes to a
    - ready drive or hot spare drive.
/ADP /C<n> - Provides the specified Controller Info.
    - Where <n> specifies the Adapter number.
/PDINFO /C<n> - Provides the PD info of the specified Controller.
Where <n> specifies the Adapter number.
/FREEARR /A<m>:d1,d2,dn.. - Provides the Free Array Info.
    - Where <m> specifies the Raid mode user needs to create.
    - Where d1, d2, dn... are the drive list.
/SETADP /C<n> [/r<val> ] [/n<val>] [/bios<-on/off>]
    [/s<-on/off>] [/f<-on/off>] [/cc<val>] [/ars<-on/off>]
    - [/boot<val>] [/arb<-on/off>]
    - Allows user to set values to adapter enquiry information.
    - /C<n> is the controller index.

```



- /r<val> set the rebuild rate of specified controllers.
- <val> should be 0-100
- /bios<-on/off> to enable/disable bios.
- /s<-on/off> is a bios stop on error setting.
- /n<val> sets the action if a new device is found.
- <val> 0: Auto-config as Spare or R0 (depend on RAID level)
- 1: Force into ^M.
- /f<-on/off> enable/disable fast initialization
- /cc<val> set action to be taken during check consistency operation.
- <val> 0: Fix/Report, 1:Report only.
- /ars<-on/off> enable/disable auto resume feature.
- /boot<val> set the bootable logical drive. <val> = LD number.
- /arb<-on/off> enable/disable auto rebuild feature.

/SETPD /C<n> [/wc<-on/off> ] [/ra<-on/off>]

- Allows user to set values to Physical drive enquiry information.
- /C<n> is the controller index.
- /wc<-on/off> enable/disable disk write cache.
- /ra<-on/off> enable/disable disk read ahead.

/DEL /l<n1> [/l<n2>] ... - Perform Ld deletion on the specified LDs.

- Where n1, n2,... specifies LD numbers.

/SETFACT /C<n> - Set the factory default.

- /C<n> is the controller index.

## Return Codes

Exit code returned can be modified using qualifier /Q.

When /Q is specified, config. related commands return ARRAY COUNT on SUCCESS. In all other cases ERROR LEVEL is returned

All the other options except /S, /F, /B, /@ & /W will have default arg as zero.

Error Codes :

6h - Array not formed due to inadequate drives

- This error occurs if adequate drives are not present
- for raid to be configured.

0Ah - Hard disk(s) not found

- This error occurs when required hard disks are not found

0Bh - Raid controller timed out on read/write

Possible reasons are damaged HDD, invalid drive id specified

0Dh - Error Opening Array Information file

- Configuration file specified could not be opened

0Eh - Array information file format incorrect

Possible reasons, missing fields/wrong information in file

10h - Error Opening Log file

- Log file name/path may be wrong/invalid

12h - Array not formed as required drives not found

- This error occurs if required drives are not found

13h - Array not formed, excess drives found

- This error occurs when more than 2 drives present in /! command

16h - Invalid configuration present

- This error occurs when no raid config. is present and
- /L command is issued

17h - Configuration mismatch in drives

- This error occurs when the config. info in all drives  
do not match. Occurs with /T command

18h - Error reading file

19h - Error writing file

21h - Physical Drive Rebuild Failed

22h - Check Consistency Failed

23h - No FreeArray Info Available

24h - Physical Drive Rebuild Aborted

25h - Check Consistency Aborted

26h - Ld Initialization Aborted

27h - Failed to connect spy

28h - Logical drive deletion failed

1Ah - Warning!! Truncating file

- This error occurs when the file dumped to a sector is  
greater than 512 bytes

1Bh - Raid BIOS not found

- Occurs with /V command if raid bios is not found

1Ch - Feature not available in IDE drive

- Error occurs if the /P or /M command is used on drives
- that do not support this feature, or invalid params. are specified

Other error codes are reserved

## Script Examples

```
#Returns information about the RAID adapter
hyperwin /ADP /C0
```

```
#Clears previous RAID volumes and configuration information and
creates RAID1 on ports #0 and #1 without asking for user input.
%deploydir%\hyperwin /AM:00,10 /S
```

```
#Initialize Volume 0
%deploydir%\hyperwin /INIT /10
```

```
#Set Disk Cache options for Port#0 and Port #1
%deploydir%\hyperwin /SETPD /C0 /wc-on /ra-on
%deploydir%\hyperwin /SETPD /C1 /wc-on /ra-on
```

# persistVariable

This tool is used to store and retrieve a value from non-volatile storage. This tool can be used to track script execution status across system reboots.

## Supported Intel® Platforms

- Intel Server Platform SR1530AH
- Intel Server Platform SR2500AL
- Intel IP Network Server NSW1U
- Intel Carrier Grade Server TIGW1U
- Intel IP Network Server NSC2U
- Intel Carrier Grade Server TIGH2U

## Supported Environments

- Red Hat Enterprise Linux 4, Update 4 (32-bit x86 Edition only)
- Windows Server 2003 SP1 (32-bit x86 Edition only)
- Windows Preinstallation Environment 2005 (32-bit x86 Edition only)

## Command Line Syntax

```
persistVariable {read Name | update Name=Value [Name=Value] [...] | verify Name=Value | delete Name | /?}
```

*Name*            A string of no more than 8 characters.

*Value*           A hexadecimal number between 0 and 7F.

## Return Codes

Read return codes:

0–127	The command completed successfully and returned the value of the variable.
-1	The environment variable has been reset or no value has been stored.

Update return codes:

0	The command completed successfully.
-1	The command failed because the wrong syntax was used for name=value, or the value as invalid.

Verify return codes:

0	The command completed successfully and the value specified is equal to the value set for the named environment variable.
-1	The command failed for one of the following reasons: Name is not set; the value specified is not equal to the actual value; or the value specified is invalid.

Delete return codes:

0	The command completed successfully.
---	-------------------------------------

## Script Examples

Example 1:

```
persistVariable update STATE=8
```

Example 2:

```
PHASE=`persistVariable read STATE`
if $PHASE = -1    # Default
    FlashMyBIOS
    persistVariable update STATE=1
    reboot
elif $PHASE = 1
    ConfigureMyLAN
    persistVariable update STATE=2
    reboot PXE
fi
```

# iFlash32

The iFlash32 utility is used to update the system BIOS, or to get information about the current system BIOS or the BIOS update package.

## Supported Intel® Platforms

- Intel Server Platform SR1530AH (BIOS updates only)
- Intel Server Platform SR2500AL
- Intel® IP Network Server NSW1U
- Intel® Carrier Grade Server TIGW1U
- Intel® IP Network Server NSC2U
- Intel® Carrier Grade Server TIGH2U

## Supported Environments

- Windows Preinstallation Environment 2005 (32-bit x86 Edition only)

## Command Line Syntax

```
iFlash32 [/h | /?] [ {filename /u [/ni] [/r]} | [filename] /i]
```

<i>filename</i>	Name of the binary file used for the update. The file path can be specified with the file name. There is no default file name or file extension. Either the "/u" or the "/i" option must also be used when specifying a file name.
/h or /?	Displays command line help. When this option is used, any other options on the command line are ignored.
/u	Update system BIOS. A binary file name must be specified with this option.
/i	Display the current BIOS version of the system. If a binary file is specified with this option, this option displays the long description contained in the file header, see Appendix A. This option is not valid with any other options.
/ni	Non-Interactive update mode. The utility displays only progress and error messages and disables prompting the user for input. This option is only supported in conjunction with the "/u" option.
/r	When this option is used, iflash32 will automatically reset the system five seconds after the update is successfully completed. This option is only supported in conjunction with the "/u" option. If an update is performed and this option is not specified, a message will be displayed stating that a manual reset must be done by the user in order for the update to take effect.

**Note:** The iFlash32 utility does not support "split capsule" BIOS files.

## Return Codes

0	Successful termination.
1	Invalid invocation.
2	File was not found.
3	Unable to read file.
4	File is invalid.
5	BIOS interface failed – this error can occur when reading or writing the BIOS or when the BIOS returns an error.
6	Utility failed to initialize.
7	Unknown error.

## Script Examples

The following command updates the BIOS from the file named “harwich7.cap”. Following the update, the user must manually reset the machine before the update will take effect.

```
C:\>iflash32 /u /r /ni harwich7.cap
IFlash32 BIOS Update Utility Ver 1.0
Copyright (c) Intel Corportation 2004

Reading file "harwich7.cap"...Done!
Examining file...Done!
File description: Intel(c)SHW4 BIOS Capsule

Updating the system BIOS...Done!
The system BIOS has been updated successfully.

The system will automatically reset in 2 seconds...
```

A sample command to perform the BIOS flash for the Intel Server Board S5000PAL:

```
iflash32 /ni /u \fw-s5\R0079.cap
```

## Additional Documentation

*User Guide for IFlash32 Utility*

# Intel® One-Boot Flash Update

The Intel® One-Boot Flash Update (Intel OFU) Utility may be used to update one or more of the following firmware components on Intel server platforms:

- System BIOS
- Baseboard Management Controller (BMC)
- Hot-swap Controller (HSC)
- Field Replaceable Unit (FRU)
- Sensor Data Record (SDR)
- Intel Local Control Panel

## Supported Intel® Platforms

- Intel Server Platform SR1530AH (BIOS updates only)
- Intel Server Platform SR2500AL
- Intel® IP Network Server NSW1U
- Intel® Carrier Grade Server TIGW1U
- Intel® IP Network Server NSC2U
- Intel® Carrier Grade Server TIGH2U

## Supported Environments

- Red Hat Enterprise Linux 4, Update 4 (32-bit x86 Edition only)
- Windows Server 2003 SP1 (32-bit x86 Edition only)

## Command Line Syntax

**flashupdt** [-i] [-u <URL or path>] [-c] [-h|?]

Updates the System BIOS or firmware on the local server with the System BIOS or firmware specified in the Intel OFU configuration file provided with the update package.

- [–i] Displays the version information for the currently running System BIOS, BMC, and SDR. If the –i option is specified with the –u option, the utility displays the version information of the update package files.
- [–u] Performs the System BIOS and firmware update. The <URL or path> parameter specifies the location where the files required for the update are located. The value of <URL or path> can be a local file system path, an FTP server, or an HTTP server. Examples of using the –u option:
  - u Specifies the current local directory.
  - u http://<IP address or URL>/<path> Specifies an HTTP server.
  - u ftp://<login:password>@<server name or IP address>/<path> Specifies an FTP server.If –u is used in conjunction with –i, no update is performed. Only the package



information is displayed.

[–c] Cancels all pending update operations of the BIOS, BMC and SDR that were performed using the utility. The utility resets the internal flags in the BIOS, BMC and SDR to cancel the update operation, whether there is one or not. FRU updates take effect immediately.

[–h|?] Displays command line help information.

**Note:** The user must have Administrator or Root privileges.

## Return Codes

0	Successful termination
1	Invalid invocation or unknown command line argument
2	File was not found
3	Unable to read a file
4	The file update package is incompatible with the target server
5	A file in the update package is invalid or unsupported
6	Firmware interface failure (an error occurred when reading or writing to the BMC, setting the update notification, or updating the BMC, FRU, HSC, Intel Local Control Panel, or SDR)
7	BIOS interface failure (an error occurred when reading the BIOS ID, setting the update notification, or updating the System BIOS)
8	Insufficient rights (the user must have Administrator or root rights)
9	Utility is already running in another process
10	Utility initialization failed

**Note:** the update configuration file (.cfg) may use the ERRORLEVEL command to override these values.

## Script Examples

```
flashupdt -i
flashupdt -u ftp://ftp.example.com/Updates/ServerName
flashupdt -u "ftp://ftp.example.com/Updates/Server Name"
flashupdt -u ftp://Kevin:87w09@ftp.example.com/Updates/ServerName
```

For Windows:

```
flashupdt -u c:\Updates\ServerName
```

For Linux:

```
flashupdt -u /Updates/ServerName
```

## Usage Notes

The BMC boot block, Intel Local Control Panel, HSC, and FRU updates takes effect immediately when the utility executes. The System BIOS, BMC Op Code, SDR, and BMC Platform Information Area (PIA) updates are programmed into their respective secondary flash areas and the utility sets an internal flag in the BIOS and BMC to indicate that the

update occurred. After a system reset, the newer version of the System BIOS, BMC, and SDRs are validated and activated.

This utility can be executed remotely through a secure network connection using a Telnet Client and Terminal Services in Windows or using a Telnet Client and Remote Shell under Linux.

## **Additional Documentation**

*Intel® One Boot Flash Update Utility User Guide*

# Ipmitool

This utility is used to configure the baseboard management controller (BMC). This is an Open Source tool and is available from <http://ipmitool.sourceforge.net/> and is also in several Linux distributions.

## Supported Intel® Platforms

- Intel Server Platform SR2500AL
- Intel® IP Network Server NSW1U
- Intel® Carrier Grade Server TIGW1U
- Intel® IP Network Server NSC2U
- Intel® Carrier Grade Server TIGH2U

## Supported Environments

- Red Hat Enterprise Linux 4, Update 4 (32-bit x86 Edition only)
- Windows Server 2003 SP1 (32-bit x86 Edition only)
- Windows Preinstallation Environment 2005 (32-bit x86 Edition only)

## Command Line Syntax

Use the following commands to get specific help:

Ipmitool alarms -?

Ipmitool health -?

Ipmitool fru -?

Ipmitool getevt -?

Ipmitool reset -?

Ipmitool cmd -?

Ipmitool lan -?

Ipmitool serial -?

Ipmitool sensor -?

Ipmitool sel -?

Ipmitool sol -?

Ipmitool wdt -?

For syntax information, refer to the User Guide distributed with the tool, or use the online help for each command.

## Return Codes

0	Success
1	Error
Other	See output messages

## Script Examples

```
ipmiutil serial -c -B 115.2k -u newuser -p newpass
ipmiutil lan -e -I 192.168.1.2 -u newuser -p newpass
ipmiutil fru
```

## Additional Documentation

Refer to the *IPMIUtil User Guide* (distributed with the tool) for the latest information on this utility.

# Log Manager Utility

The Log Manager is a collection of three tools (addLogRecord, getLogSeverity, setLogLevel) to add, set, and get entries to a log file.

The Log Manager tool allows you to do the following:

- Append a record to the log
- Set the minimum severity level for logging
- Get the severity level of the log record with the highest severity

**Note:** this log file is created by the developer for the deployment script log file. This is not the System Log.

## Supported Intel® Platforms

- Intel Server Platform SR1530AH
- Intel Server Platform SR2500AL
- Intel IP Network Server NSW1U
- Intel Carrier Grade Server TIGW1U
- Intel IP Network Server NSC2U
- Intel Carrier Grade Server TIGH2U

## Supported Environments

- Red Hat Enterprise Linux 4, Update 4 (32-bit x86 Edition only)
- Windows Server 2003 SP1 (32-bit x86 Edition only)
- Windows Preinstallation Environment 2005 (32-bit x86 Edition only)

## Command Line Syntax

```
addLogRecord Source {CRITICAL | ERROR | WARNING | INFO | DEBUG}
```

where

*Source* is a string that identifies the log record

*Message* is a string that is the text to be logged

```
getLogSeverity
```

```
setLogLevel {CRITICAL | ERROR | WARNING | INFO | DEBUG}
```

## Return Codes

addLogRecord Return Codes:

0      Successful

-1     Not successful

SetLogLevel Return Codes:

0      Successful

-1     Not successful

getLogSeverity Return Codes:

0      No log records found

1      DEBUG

2      INFO

3      WARNING

4      ERROR

5      CRITICAL

## Script Examples

```
addLogRecord EEUPDATE INFO "mac address is 0015170BCDEE for 8086-105E Intel(R)  
PRO/1000 PT Dual Port Server Adapter"
```

```
addLogRecord EEUPDATE ERROR "e1000 driver failed to load, return value 3"
```

```
getLogSeverity
```

```
setLogLevel ERROR
```

# nicSpeed

This tool is a VBScript that is used to set the speed and mode of the system NICs under Windows.

## Supported Intel® Platforms

- Intel Server Platform SR1530AH
- Intel Server Platform SR2500AL
- Intel IP Network Server NSW1U
- Intel Carrier Grade Server TIGW1U
- Intel IP Network Server NSC2U
- Intel Carrier Grade Server TIGH2U

## Supported Environments

- Windows Preinstallation Environment 2005 (32-bit x86 Edition only)
- Windows Server 2003 SP1 (32-bit x86 Edition only)

## Command Line Syntax

```
nicSpeed [-i NIC_index] | [-s Speed_index] | -?
```

where:

-i                Shows information for Intel network devices

-s                Set the NIC speed

*NIC\_index*      The NIC number, starting at 0.

*Speed\_index*    The NIC Speed index number (shown by the -i command)

-?                Displays help for the command

Used without parameters, nicSpeed will display all IP-enabled Intel NIC devices by index number.

## Return Codes

0                Successful

-1                Failure (invalid *NIC\_index* or *Speed\_index*)

## Script Examples

```
nicSpeed //nologo -i 7
```

# RaidCfg32

This utility configures the Intel MATRIX RAID.

## Supported Intel® Platforms

- Intel Server Platform SR1530AH

## Supported Environments

- Windows Server 2003 SP1 (32-bit x86 Edition only)
- Windows Preinstallation Environment 2005 (32-bit x86 Edition only)

## Command Line Syntax

RaidCfg32.exe [/?] [/Y] [/Q] [/RD] [/SPARE:drive] [/C:vol\_name] [/SS:strip\_size]  
[/L:raid\_level] [/S:vol\_size] [/DS:disk\_ports] [/D:vol\_name] [/X] [/I] [/P] [/ST] [/STD]  
[/STV] [/F:vol\_name] [/CnG] [/Sync] [/M]

/?	Displays Help Screen. Other options ignored.
/Y	Suppress any user input. Used with options /C, /D, /X, & /F.
/Q	Quiet mode / No output. Should not be used with status commands.

COMMANDS (Use only one at a time):

/RD	Inquire if the raid driver is running
/SPARE	Set a drive to be a spare
/C	Create a volume with the specified name. /S, /DS, /SS, & /L can be specified along with /C.
/SS	Specify strip size in KB. Only valid with /C
/L	Specify RAID Level (0, 1, 10, or 5). Only valid with /C
/S	Specify volume size in GB or percentage if a '%' is appended. Percentage must be between 1-100. Only valid with /C
/DS	Selects the disks to be used in the creation of volume. List should be delimited by spaces.
/D	Delete Volume with specified name.
/X	Remove all metadata from all disks. Use with /DS to delete metadata



from selected disks.

/I	Display All Drive/Volume/Array Information. /P can be specified.
/P	Pause display between sections. Only valid with /I.
/ST	Display Volume/RAID/Disk Status.
/STD	Display delimited Disk Status Port,Model,SerialNumber,FirmwareVersion,Array,Status,Size,Free,Type
/STV	Display delimited Volume Status Index,Level,StripSize,Size,Status,Bootable,Array,Name
/F	Repair failed RAID0 Volume.
/CnG	Create a recovery volume. Only valid with /C.
/Sync	Set the sync type for the the recovery volume. Default is auto. Only valid with /CnG
/M	Specify the disk that is to be the master disk when creating recovery volume.

# Reboot

This utility reboots the system.

## Supported Intel® Platforms

- Intel Server Platform SR1530AH
- Intel Server Platform SR2500AL
- Intel IP Network Server NSW1U
- Intel Carrier Grade Server TIGW1U
- Intel IP Network Server NSC2U
- Intel Carrier Grade Server TIGH2U

## Supported Environments

- Red Hat Enterprise Linux 4, Update 4 (32-bit x86 Edition only)
- Windows Server 2003 SP1 (32-bit x86 Edition only)
- Windows Preinstallation Environment 2005 (32-bit x86 Edition only)

## Command Line Syntax

**reboot**

## Return Codes

- 0 Successful completion.
- 1 Generic or unspecified error.
- 2 Invalid or excess arguments.
- 3 Unimplemented feature.
- 4 User had insufficient privileges.
- 5 Program is not installed.
- 6 Program is not configured.
- 7 Program is not running.

## Script Example

```
PHASE=`persistVariable read STATE`  
if $PHASE = -1    # Default  
    FlashMyBIOS  
    persistVariable update STATE=1
```

```
        reboot
elif $PHASE = 1
    ConfigureMyLAN
    persistVariable update STATE=2
    reboot PXE
fi
```

# SCSI RAS Tools for Linux

This is an Open Source collection of tools to enhance the Reliability, Availability, and Serviceability of SCSI/SAS disks, and Linux Root Disk Mirroring. The Linux tools include: sgdiskmon, sgraidmon, sgdefects, sgdiskfl, sgmode, and sgdiag.

For Windows Server and Windows Preboot Execution Environment 2005 tools, refer to the separate pages for: slist, sdiskfw, and sdiskmon.

## Supported Intel® Platforms

- Intel Server Platform SR2500AL
- Intel IP Network Server NSW1U
- Intel Carrier Grade Server TIGW1U
- Intel IP Network Server NSC2U
- Intel Carrier Grade Server TIGH2U

## Supported Environments

- Red Hat Enterprise Linux 4, Update 4 (32-bit x86 Edition only)

## Command Line Syntax

Refer to the documentation provided with the tools. The Linux version is available at <http://scsirastools.sourceforge.net/>.

## Script Examples

```
# wcache: 0=disable, 1=enable
wcache=0
logfile=$DeploymentStore/cfgdisk.log
# set disk write cache on all disks
sgmode -a -w$wcache >>$logfile 2>&1
```

## Additional Documentation

Refer to the documentation provided with the tools.

# SCSI RAS Tools for Windows

This is an Open Source collection of tools to enhance the Reliability, Availability, and Serviceability of SCSI/SAS disks, and Linux Root Disk Mirroring.

This section lists the tools that run under Windows Server and Windows Preboot Execution Environment 2005. These tools include: slist, sdiskfw, and sdiskmon.

For Linux see SCSI RAS Tools for Linux. The Linux tools include: sgdiskmon, sgraidmon, sgdefects, sgdiskfl, sgmode, and sgdiag.

## Supported Intel® Platforms

- Intel Server Platform SR2500AL
- Intel IP Network Server NSW1U
- Intel Carrier Grade Server TIGW1U
- Intel IP Network Server NSC2U
- Intel Carrier Grade Server TIGH2U

## Supported Environments

- Windows Server 2003 SP1 (32-bit x86 Edition only)
- Windows Preinstallation Environment 2005 (32-bit x86 Edition only)

## Command Line Syntax

### SLIST Syntax

Used to get/set SCSI mode pages, and optionally get defect lists.

```
slist [-ademsvy]
```

The slist command without options will display a list of available disks and their properties.

The options are:

- |         |  |
|---------|--|
| -a      | show all automatically                   |
| -d      | show drive defects                       |
| -e      | don't write to slist.log file            |
| -m      | don't show mode pages                    |
| -s      | set mode pages from *.mdf file           |
| -v      | show verbose and debug messages          |
| -w{0 1} | set the WriteCache (0=disable, 1=enable) |

-y      yes, update the capacity, if different

## SDISKFW

Used to flash SCSI disk/tape firmware.

```
sdiskfw [-e -i disknum -m diskmodel -f image_file -t seconds_delay -v]
```

-m	Automatically write to all of this model
<i>diskmodel</i>	The disk model. The first 8 characters in the Description shown in slist is the Vendor ID string; the next 16 characters of the description is the model/product ID string. Use the model/product ID string.
-i	Automatically do only this drive number
<i>disknum</i>	The physical disk number, 0-31, displayed by slist
-f	Specify filename of the firmware image
<i>Image_file</i>	The name of the image file. You may specify any file name. The file name may include a path. If no path is specified then the current directory is the default path.
-t	Time to delay after download, in seconds. The default value is 10 seconds.
-e	Do not write to any files
-c	Crash afterwards if doing disk 0 (requires crashd.sys)
-v	Verbose debug messages
-x	Issue Stop Unit command
-y	Issue Start Unit command

### SDISKFW Error Codes

0x0XXX	Errors before or during download command
0x1XXX	Errors after download with TestUnitReady
0x2XXX	Errors after download with make online
0x3XXX	Errors after download with inquiry

## SDISKMON

Used to monitor SCSI disks for a failure, and set disk fault LED.

```
sdiskmon [-a -b -e {-t n} -v]
```

-a	Automatically repeat
-b	Run in background in a repeat loop

- e Don't write to sdiskmon.log file
- r Run once only, don't repeat
- t *n* Time to sleep between loops, the default is 5 seconds.
- v Show verbose and debug messages

# Syscfg

This utility is used to configure the BIOS and BMC settings.

Syscfg has a save and restore feature that allows you to capture a server configuration and replicate it on other servers or reset the source server. Some settings, however, are not saved. The static IP Address assigned by a DHCP server, BIOS boot order, date and time, and other dynamic BIOS settings are not saved or restored.

## Supported Intel® Platforms

- Intel Server Platform SR1530AH (BIOS settings only)
- Intel Server Platform SR2500AL
- Intel IP Network Server NSW1U
- Intel Carrier Grade Server TIGW1U
- Intel IP Network Server NSC2U
- Intel Carrier Grade Server TIGH2U

## Supported Environments

- Red Hat Enterprise Linux 4, Update 4 (32-bit x86 Edition only)
- Windows Server 2003 SP1 (32-bit x86 Edition only)
- Windows Preinstallation Environment 2005 (32-bit x86 Edition only)

## Command Line Syntax

Refer to the *Intel® System Configuration Utility User Guide* for syntax and usage information.

## Script Examples

```
syscfgd=/opt/deploy/Linux/syscfg
cd $syscfgd
# dump the BIOS parameters
./syscfg /d BIOS

# save the BIOS configuration
rm -f $cfgfile 2>/dev/null
./syscfg /s $cfgfile /b

# restore the BIOS configuration
./syscfg /r $cfgfile /b
```

## Additional Documentation

*Intel® System Configuration Utility User Guide*



# System Information Manager

The System Information Manager is a collection of tools (addProperty, queryProperty, countProperty) that are used to add a record to an inventory file, query, or count the records in the inventory file.

This tool has the following capabilities:

- Query the inventory file created by addProperty.
- Add or modify a record in an inventory file.
- Count the various system properties that have been added by the addProperty command.
- List the properties available for use with the other tools.

## Supported Intel® Platforms

- Intel Server Platform SR1530AH
- Intel Server Platform SR2500AL
- Intel IP Network Server NSW1U
- Intel Carrier Grade Server TIGW1U
- Intel IP Network Server NSC2U
- Intel Carrier Grade Server TIGH2U

## Supported Environments

- Red Hat Enterprise Linux 4, Update 4 (32-bit x86 Edition only)
- Windows Server 2003 SP1 (32-bit x86 Edition only)
- Windows Preinstallation Environment 2005 (32-bit x86 Edition only)

## Command Line Syntax

Use addProperty to add new properties for existing and new sources. Use queryProperty to query the Name/Value pairs for specific records. Use countProperty to count the number of instances of a specific property. Use listProperties to discover the parameters used with the query Property command.

**addProperty** { *Source Name Value instance="Number" [Attribute="Value" [...]]* } | /?

**countProperty** { *Source Name [Attribute="Value" [...]]* } | /?

**queryProperty** { *Source Name { - | Value } [instance="Number"] [Attribute="Value" [...]]* } | /?

Where:

<i>Source</i>	<p>The source of the system property:</p> <p>SMBIOS FRU SDR AMT Disk IntelBIOSID Ethernet ?</p> <p>or a source defined by the user with the addProperty command. When "?" is used as the source, all sources and their qualifying attributes are displayed.</p>
<i>Name</i>	<p>The name of the system property or "?". When "?" is used as the name, all properties with their qualifying attributes are displayed.</p>
<i>Value   -</i>	<p>The value of the system property. For the queryProperty command, if this parameter is a hyphen ("-"), then the value string or strings matching the identifying attributes will be sent to "stdout."</p>
<i>Attribute=" Value"</i>	<p>Additional attributes, if required by the source. For example, the SMBIOS table source requires additional attributes. Multiple Attribute/Value pairs must be separated by a space.</p>
<b>instance="Number"</b>	<p>The instance of the system property. For example, if the SMBIOS table contains more than one instance of a property name, you can use the instance attribute to specify the instance of the record you want. Number must be 1 or greater.</p>
<i>/?</i>	<p>Provides help on the utility.</p>

#### **Notes**

- The Intel 5000 Series Chipsets Server Board Family uses the following sources: SMBIOS, FRU, SDR, DISK, and Ethernet. Intel® 3000 Series Chipsets Server Board Family uses the SMBIOS, AMT, DISK, and Ethernet sources.
- The SMBIOS on-board device information records typically contain multiple instances (only one instance is shown in the following example).

## Source, Attributes, and Name Parameters

Source	Attributes	Name	Typical Value
SMBIOS	info="System Information" type="1"	Manufacturer	Intel
SMBIOS	info="System Information" type="1"	Product Name	S5000PHB
SMBIOS	info="System Information" type="1"	Version	Not Specified
SMBIOS	info="System Information" type="1"	Serial Number	CFTF7080015
SMBIOS	info="System Information" type="1"	UUID	CDB6BC01-C459-11DB-B929-000E0CC6D48A

SMBIOS	info="System Information" type="1"	Wake-up Type	Power Switch
SMBIOS	info="System Information" type="1"	SKU Number	NSWD0201W
SMBIOS	info="System Information" type="1"	Family	Not Specified
SMBIOS	info="Base Board Information" type="2"	Manufacturer	Intel
SMBIOS	info="Base Board Information" type="2"	Product Name	S5000PHB
SMBIOS	info="Base Board Information" type="2"	Version	FRU Ver 0.05
SMBIOS	info="Base Board Information" type="2"	Serial Number	CFTW70400036
SMBIOS	info="Base Board Information" type="2"	Asset Tag	SilFR
SMBIOS	info="Base Board Information" type="2"	Location In Chassis	Not Specified
SMBIOS	info="Base Board Information" type="2"	Chassis Handle	0x0000
SMBIOS	info="Base Board Information" type="2"	Type	Motherboard
SMBIOS	info="Base Board Information" type="2"	Contained Object Handles	0
SMBIOS	info="Chassis Information" type="3"	Manufacturer	NSW1U
SMBIOS	info="Chassis Information" type="3"	Type	Rack Mount Chassis
SMBIOS	info="Chassis Information" type="3"	Lock	Not Present
SMBIOS	info="Chassis Information" type="3"	Version	NSW1U
SMBIOS	info="Chassis Information" type="3"	Serial Number	10.243.43.43
SMBIOS	info="Chassis Information" type="3"	Asset Tag	SilFR
SMBIOS	info="Chassis Information" type="3"	Boot-up State	Safe
SMBIOS	info="Chassis Information" type="3"	Power Supply State	Safe
SMBIOS	info="Chassis Information" type="3"	Thermal State	Safe
SMBIOS	info="Chassis Information" type="3"	Security Status	Unknown
SMBIOS	info="Chassis Information" type="3"	OEM Information	0x81581CF8
SMBIOS	info="Chassis Information" type="3"	Height	1 U
SMBIOS	info="Chassis Information" type="3"	Number Of Power Cords	1
SMBIOS	info="Chassis Information" type="3"	Contained Elements	0
SMBIOS	info="BIOS Information" type="0"	Vendor	Intel Corporation
SMBIOS	info="BIOS Information" type="0"	Version	S5000.86B.07.00.D393.080920071743
SMBIOS	info="BIOS Information" type="0"	Release Date	8/9/2007
SMBIOS	info="BIOS Information" type="0"	Address	0xE8000
SMBIOS	info="BIOS Information" type="0"	Runtime Size	96 kB
SMBIOS	info="BIOS Information" type="0"	ROM Size	4096 kB
SMBIOS	info="BIOS Information" type="0"	BIOS Revision	7
SMBIOS	info="BIOS Information" type="0"	Firmware Revision	0
SMBIOS	info="OEM Strings" type="11"	String 1	j
SMBIOS	info="OEM Strings" type="11"	String 2	j
SMBIOS	info="OEM Strings" type="11"	<String 3 - String 55>	j
SMBIOS	info="OEM Strings" type="11"	String 56	j
SMBIOS	info="OEM Strings" type="11"	String 57	j
SMBIOS	info="On Board Device Information" instance="1" type="10"	Type	Video
SMBIOS	info="On Board Device Information" instance="1" type="10"	Status	Enabled
SMBIOS	info="On Board Device Information" instance="1" type="10"	Description	ATI Rage XL
SMBIOS	info="Memory Device" instance="1" type="17"	Array Handle	0x0022

SMBIOS	info="Memory Device" instance="1" type="17"	Error Information Handle	Not Provided
SMBIOS	info="Memory Device" instance="1" type="17"	Total Width	64 bits
SMBIOS	info="Memory Device" instance="1" type="17"	Data Width	64 bits
SMBIOS	info="Memory Device" instance="1" type="17"	Size	512 MB
SMBIOS	info="Memory Device" instance="1" type="17"	Form Factor	DIMM
SMBIOS	info="Memory Device" instance="1" type="17"	Set	None
SMBIOS	info="Memory Device" instance="1" type="17"	Locator	J8J1
SMBIOS	info="Memory Device" instance="1" type="17"	Bank Locator	CHAN A DIMM 1
SMBIOS	info="Memory Device" instance="1" type="17"	Type	DDR2
SMBIOS	info="Memory Device" instance="1" type="17"	Type Detail	Synchronous
SMBIOS	info="Memory Device" instance="1" type="17"	Speed	667 MHz (1.5 ns)
SMBIOS	info="Memory Device" instance="1" type="17"	Manufacturer	0x0000000000000000
SMBIOS	info="Memory Device" instance="1" type="17"	Serial Number	0xFFFFFFFF
SMBIOS	info="Memory Device" instance="1" type="17"	Asset Tag	Unknown
SMBIOS	info="Memory Device" instance="1" type="17"	Part Number	0x00000000000000000000000000000000
SMBIOS	info="Processor Information" type="4"	Socket Designation	CPU1
SMBIOS	info="Processor Information" type="4"	Type	Central Processor
SMBIOS	info="Processor Information" type="4"	Family	Xeon
SMBIOS	info="Processor Information" type="4"	Manufacturer	Intel(R) Corporation
SMBIOS	info="Processor Information" type="4"	ID	F6 06 00 00 FF FB EB BF
SMBIOS	info="Processor Information" type="4"	Signature	Type 0, Family 6, Model 15, Stepping 6
SMBIOS	info="Processor Information" type="4"	Version	Intel(R) Xeon(R) CPU 5140 @ 2.33GHz
SMBIOS	info="Processor Information" type="4"	Voltage	1.2 V
SMBIOS	info="Processor Information" type="4"	External Clock	1333 MHz
SMBIOS	info="Processor Information" type="4"	Max Speed	2333 MHz
SMBIOS	info="Processor Information" type="4"	Current Speed	2333 MHz
SMBIOS	info="Processor Information" type="4"	Status	Populated, Enabled
SMBIOS	info="Processor Information" type="4"	Upgrade	ZIF Socket
SMBIOS	info="Processor Information" type="4"	L1 Cache Handle	0x0037
SMBIOS	info="Processor Information" type="4"	L2 Cache Handle	0x0036
SMBIOS	info="Processor Information" type="4"	L3 Cache Handle	Not Provided
SMBIOS	info="Processor Information" type="4"	Serial Number	Not Specified
SMBIOS	info="Processor Information" type="4"	Asset Tag	Not Specified
SMBIOS	info="Processor Information" type="4"	Part Number	Not Specified

SMBIOS	info="Processor Information" type="4"	Core Count	2
SMBIOS	info="Processor Information" type="4"	Core Enabled	2
SMBIOS	info="Processor Information" type="4"	Thread Count	2
FRU	info="Front Panel FRU"	Board Manufacturer	Intel
FRU	info="Front Panel FRU"	Board Product Name	EFP Board
FRU	info="Front Panel FRU"	Board Serial Number	CFTW70400316
FRU	info="Front Panel FRU"	Board Part Number	D55392-502
FRU	info="Front Panel FRU"	Board FRU File ID	FRU Ver 0.01
FRU	info="Power Dist FRU"	Product Manufacturer	DELTA
FRU	info="Power Dist FRU"	Product Name	AC-061 A
FRU	info="Power Dist FRU"	Product Part Number	D39926-006
FRU	info="Power Dist FRU"	Product Version	S5
FRU	info="Power Dist FRU"	Product Serial Num	DLD0703000587
FRU	info="Pwr Supply 1 FRU"	Product Manufacturer	DELTA
FRU	info="Pwr Supply 1 FRU"	Product Name	DPS-450KBA
FRU	info="Pwr Supply 1 FRU"	Product Part Number	D40117-003
FRU	info="Pwr Supply 1 FRU"	Product Version	S2
FRU	info="Pwr Supply 1 FRU"	Product Serial Num	DLC0630000116
FRU	info="Mainboard FRU"	Chassis Type	Rack-Mount Chassis
FRU	info="Mainboard FRU"	Chassis Part Number	NSW1U
FRU	info="Mainboard FRU"	Chassis Serial Num	10.243.43.43
FRU	info="Mainboard FRU"	Chassis OEM Field	NSW1U
FRU	info="Mainboard FRU"	Board Manufacturer	Intel
FRU	info="Mainboard FRU"	Board Product Name	S5000PHB
FRU	info="Mainboard FRU"	Board Serial Number	CFTW70400036
FRU	info="Mainboard FRU"	Board Part Number	D40558-440
FRU	info="Mainboard FRU"	Board FRU File ID	FRU Ver 0.05
FRU	info="Mainboard FRU"	Product Manufacturer	Intel
FRU	info="Mainboard FRU"	Product Name	S5000PHB
FRU	info="Mainboard FRU"	Product Part Number	NSWD0201W
FRU	info="Mainboard FRU"	Product Serial Num	CFTF7080015
FRU	info="Mainboard FRU"	Product Asset Tag	SilFR
FRU	info="Mainboard FRU"	System GUID	cdb6bc01-c459-11db-b929-000e0cc6d48a
SDR		BMC Version	0.16
SDR		IPMI Version	2.0
SDR		File Version	12
SDR		Package Version	12
DISK	instance="0" type="SCSI"	device	/dev/sda
DISK	instance="0" type="SCSI"	Vendor	SEAGATE
DISK	instance="0" type="SCSI"	Model	HDS725050KLA360
DISK	instance="0" type="SCSI"	Serial Number	3LC01RZF
DISK	instance="0" type="SCSI"	Firmware Revision	002
DISK	instance="0" type="SCSI"	MB Capacity	35003
DISK	instance="0" type="ATA"	device	/dev/hdb
DISK	instance="0" type="ATA"	Vendor	SEAGATE
DISK	instance="0" type="ATA"	Model	HDS725050KLA360
DISK	instance="0" type="ATA"	Serial Number	KRVN66ZAJM80JD

DISK	instance="0" type="ATA"	Firmware Revision	K2AOAD1A
DISK	instance="0" type="ATA"	MB Capacity	476940
Ethernet	instance="0"	MACAddress	00:0E:0C:C6:D4:8E
Ethernet	instance="1"	MACAddress	00:0E:0C:C6:D4:8F
Ethernet	instance="2"	MACAddress	00:0E:0C:C6:D4:8A
Ethernet	instance="3"	MACAddress	00:0E:0C:C6:D4:8B
AMT	info="Active Management Technology"	DHCP Mode	2 (Shared)
AMT	info="Active Management Technology"	Local Addr	0.0.0.0
AMT	info="Active Management Technology"	Subnet Mask	0.0.0.0
AMT	info="Active Management Technology"	Gateway	0.0.0.0
AMT	info="Active Management Technology"	Primary DNS	0.0.0.0
AMT	info="Active Management Technology"	Second DNS	0.0.0.0
AMT	info="Active Management Technology"	Host Name	r1f2e1603568b7f
AMT	info="Active Management Technology"	Max Transfer Unit	4160
AMT	info="Active Management Technology"	Provisioning Mode	1 (Enterprise)
AMT	info="Active Management Technology"	Provision State	0 (PreBoot)
AMT	info="Active Management Technology"	Network State	1 (Enabled)
AMT	info="Active Management Technology"	SOL Status	1 (Enabled)
AMT	info="Active Management Technology"	IDE-R Status	1 (Enabled)
AMT	info="Active Management Technology"	BIOS version	Not available
AMT	info="Active Management Technology"	ROM version	1.32.0
AMT	info="Active Management Technology"	Flash version	1.3.11
AMT	info="Active Management Technology"	Netstack version	3.7.0
AMT	info="Active Management Technology"	AMTApps version	3.11.0
AMT	info="Active Management Technology"	Patch1 version	4.10.0
AMT	info="Active Management Technology"	Patch2 version	4.13.0
IntelBIOSID		BoardFamilyID	S3000
IntelBIOSID		OEMID	86B
IntelBIOSID		MajorRev	02
IntelBIOSID		MinorRev	00
IntelBIOSID		BuildID	0044
BIOSRAID		HWRAID	Disabled
BIOSRAID		SWRAID	Disabled
BIOSRAID		Vendor	LSI

## Return Codes

### addProperty Return Codes:

- 0 Successful completion.
- 1 Command was not completed successfully.

### For the queryProperty **Source Name** – syntax:

- 0 The command was completed with the current value(s) sent to “stdout.”
- 1 Command failed. The attributes do not match existing properties.

### For the queryProperty **Source Name Value** syntax:

- 0 The current value identified by the attributes is not the same as the value in the command.
- 1 The current value identified by the attributes is the same as the value of the command.
- 1 The attributes do not match any property.

#### **countProperty return codes:**

- 0 no properties found properties identified by the attributes.
- n* the number of properties identified by the attributes.

#### **listProperties return codes:**

- 0 the command was completed with the current Sources and Attributes sent to stdout..
- 1 the identifying attributes are in an invalid format, Source is not found, or no Sources are available (run 'prepareDeployment' to create the inventory file).

## **Script Examples**

#### **addProperty example:**

```
addProperty SMBIOS Manufacturer Intel type="1" info="System"
```

#### **queryProperty examples:**

```
queryProperty SMBIOS Manufacturer Intel type="1" info="System Information"
```

Returns 1 if the "Intel" is the current value; 0 if the "Intel" is not the current value, else -1 if no property matched the attributes.

```
queryProperty SMBIOS Manufacturer - type="1" info="System Information"
```

Returns 0 after sending "Intel" to stdout, else -1 if no property matched the attributes.

```
queryProperty SMBIOS Version -
```

Lists all the values that have "Version" as a property name in the SMBIOS record. The source must be qualified (for example, if the source is SMBIOS, add type="0") to list a single value.

```
queryProperty SMBIOS ? type="0"
```

Lists all the property names in the SMBIOS type 0 record.

#### **countProperty examples:**

```
#!/usr/bin/python
# -*- coding: latin-1 -*-

import os, sys, systemMgr
from logMgr import addLogRecord, CRITICAL, ERROR, WARNING, INFO, DEBUG
```

```

def demo():
    """
    arg 1 is generally 0, for multiple queries use the return value of
    systemMgr.openSysInfo()
    e.g.:
    tok=systemMgr.openSysInfo()
    systemMgr.countProperty(tok, ...)
    systemMgr.countProperty(tok, ...)
    systemMgr.closeSysInfo(tok)
    arg 2 is SOURCE
    arg 3 is NAME
    arg 4 is an optional dictionary of qualifying attributes
    """
    ret = systemMgr.countProperty(0, 'Ethernet', 'MACAddress', {})
    print "%d Ethernet ports found by the OS" % ret
    return ret

if __name__ == '__main__':
    sys.exit(demo())

```

```
countProperty SMBIOS Manufacturer type="4" info="Processor Information"
```

returns the number of processors.

```
countProperty Ethernet MACAddress
```

returns the number of Ethernet ports redetected by the Operating System



# Glossary

---

FRU	Field Replaceable Unit
SDR	Sensor Data Record
BMC	Baseboard Management Controller
Intel OFU	Intel One Boot Flash Update
Intel LCP	Intel Local Control Panel
PIA	Platform Information Area
HSC	Hot Swap Controller
AMT	Intel® Active Management Technology (Intel® AMT)



# Index

---

addLogRecord, 61  
addLogRecordDemo.sh,  
    addLogRecordDemo.cmd,  
    addLogRecordDemo.py, 6  
addProperty, 73  
addPropertyDemo.sh,  
    addPropertyDemo.cmd,  
    addPropertyDemo.py, 6  
amifldr\_mod.o, 7  
AMTCfg, 15  
amtcfg.py, 6  
BootDevice, 11  
cfgamt.sh, cfgamt.cmd, 6  
cgbios.sh, cgbios.cmd, 6  
cgbmc.sh, cgbmc.cmd, 6  
cfgdisk-ata.cmd, 6  
cfgdisk-ata.sh, 6  
cfgdisk-scsi.sh, cfgdisk-scsi.cmd, 6  
cfnic.cmd, cfnic.sh, 6  
CmdTool2, 17  
cmdtool2-createvolume.sh, cmdtool2-  
    createvolume.cmd, 6  
countProperty, 73  
customDeployment, 12  
deploy.cmd, 12  
DeploymentStore, 11  
DeploymentStore root folder, 11  
DeployPath, 10  
Disc-Wipe, 19  
disc-wipe, source location, 2  
dmidecode, 22  
documentation, list of, 4  
drivers  
    amifldr\_mod.o, 7  
    E1000.ko, 7  
    Flashud.sys, 7  
    Iastor.sys, 7  
    Medaide.sys, 7  
    Megasr.ko, Megasr.sys, 7  
drivers, loading, 8  
E1000.ko, 7  
environment variables, 10  
Erase, 20  
erase.sh, erase.cmd, 6  
ETHTOOL, 23  
Ethtool, source location, 2  
evaluateDeployment, 12  
flash.py, 6  
flashamt.sh, flashamt.cmd, flashamt.py, 6  
flashbios-s3.sh, flashbios-s3.cmd, 6  
flash-s5.sh, flash-s5.cmd, 6  
Flashud.sys, 7  
FRU, 24  
FRUSDR, 24  
frusdr.py, 6  
FWPIAUPD, 31  
getLogSeverity, 61  
getLogSeverityDemo.sh,  
    getLogSeverityDemo.cmd,  
    getLogSeverityDemo.py, 6  
hdparm, 39  
hdParm, source location, 2  
HyperCfg, 42  
hypercfg-createvolume.sh, 6  
Hyperwin, 47  
Iastor.sys, 7  
iflash.py, 6  
iFlash32, 54  
imb.ko, 7  
Imbdrv.sys, 7  
installation, 9  
Intel AMT, 15  
Intel e1000 driver, source location, 2  
Intel IMB driver for IPMI, source location,  
    2  
Intel® Deployment Assistant, 2  
Intel® One-Boot Flash Update, 56  
Intel® Rapid Boot Toolkit, 3  
IpmiUtil, 59  
IPMIUtil, source location, 2

LD\_LIBRARY\_PATH, 11  
 Log Manager, 61  
 Medaide.sys, 7  
 Megasr.ko, Megasr.sys, 7  
 NIC Configuration VBScript, source location, 2  
 NicSpeed, 63  
 nicSpeed.cmd, nicSpeed.vbs, 6  
 ofu.py, 6  
 operating systems, supported, 4  
 PATH, 10  
 persistVariable, 52  
 persistVariableDemo.sh, persistVariableDemo.cmd, 6  
 platforms, supported, 4  
 prepareDeployment, 12  
 PYTHONPATH, 11  
 queryProperty, 73  
 queryPropertyDemo.sh, queryPropertyDemo.cmd, queryPropertyDemo.py, 6  
 RaidCfg32, 64  
 raidcfg-createvolume.cmd, 6  
 rc.local, 11  
 Reboot, 66  
 reboot.sh, reboot.cmd, 6  
 reboot-s5.sh, 6  
 restore defaults, 20  
 rmm-info.cmd, 6  
 runtime environment setup, 10  
 SCSI RAS Tool, 68, 69  
 SCSI RAS tools, source location, 2  
 SDR, 24  
 setDeploymentEnv, 10, 12  
 setLogLevelDemo.sh, setLogLevelDemo.cmd, setLogLevelDemo.py, 6  
 setLogLevel, 61  
 SMBIOS, 24  
 SMBiosDecode, source location, 2  
 SMI driver for syscfg, source location, 2  
 Smi.ko, Smi.sys, 7  
 source code, location of, 2  
 Spy, 42, 47  
 StoreDevice, 11  
 Syscfg, 72  
 System Information Manager, 73  
 wipe disk, 20