Intel® Server Debug and Provisioning Tool

User Guide

A setup, usage and troubleshooting guide for Intel® Server Systems

Rev 2.0–0

April 2021

Delivering Breakthrough Datacenter System Innovation – Experience What's Inside!

<This page is intentionally left blank>

Document Revisions

Date	Revision	Changes
October 2020	1.4–1	• Python* 3.0 Support • Help modification
November 2020	1.4–2	• Support added for below OSes • Red Hat* Enterprise Linux* 8.x • SuSE* Linux* Enterprise Server 15 • Centos 8.x and • Ubuntu 20.04 LTS • Python* 3.x support added for below platforms • Intel(R) Server Board S2600 series (Intel(R) Xeon(R) processor E5-2600 v3/v4 product family) • Intel® Server Board (Intel® Xeon® Scalable Processor Family) • Intel® Server Board (2nd Generation Intel® Xeon® Scalable Processor Family) • Intel® Server Board (Intel® Xeon® Platinum 9200 Processor Family) • Custom_deploy update with –no_user_interaction • Update with –no_user_interaction • Get BIOS options with –no_user_interaction • sdptool_update script issue. • Error_Code added for Help message
January 2021	1.4–3	• Fixed Static scan issues • On Intel(R) Server Board S2600 series (Intel(R) Xeon(R) processor E5-2600 v3/v4 product family), • Fixed Vmedia mount failing next time after trying to mount the image using Vmedia with –no_user_interaction switch (even though not supported) • Fixed get SEL when "-" was present in the path to save the file.
March 2021	1.4–4	• Redfish Integration to fifteen existing commands (input mechanism will be same as in legacy IPMI as before however the output might slightly change depending upon availability of redfish on the target machine). • Changes in input and output mechanism for four new commands. • Some changes in the install scripts to use native package managers for installation, update and uninstallation. • Added new exit error code for invalid length for FRU fields.
April 2021	2.0–0	• New Platform M70KLP support added. • storageinfo and nicinfo added which will provide information regarding storage and network devices.

Disclaimers

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software, or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at intel.com.

You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

The products described 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.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting www.intel.com/design/literature.htm.

Intel and the Intel logo are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries.

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

Table of Contents

List of Tables

Table of Figures

1.   Introduction

The Intel® Server Debug and Provisioning Tool (Intel® SDP Tool) is a single-server tool to debug and provision Intel® Server Boards and Systems through the BMC Out-of-band.

Intel® SDPTool is designed to work with the following Intel® Server Board families:

- Intel® Server Board S2600WT/S2600WTR family
- Intel® Server Board S2600KP/S2600KPR family
- Intel® Server Board S2600TP/S2600TPR family
- Intel® Server Board S2600CW/S2600CWR family
- Intel® Server Board S2600WF/S2600WFR family
- Intel® Server Board S2600ST/S2600STR family
- Intel® Server Board S2600BP/S2600BPR family
- Intel® Server Board S9200WK family
- Intel® Server Board D50TNP family
- Intel® Server Board M50CYP family
- Intel® Server Board M70KLP family

1.1   Document Scope

The purpose of this user guide is to help system/server administrators install and use the Intel® Server Debug and Provisioning Tool (Intel® SDP Tool). This guide provides information on the features and benefits of Intel® SDP Tool, as well as software requirements, supported operating systems and platforms. This guide also explains the installation and uninstallation process.

Note: Refer to the *Intel® Server Debug and Provisioning Tool Release Notes* for known issues on platforms and during the installation.

1.2   System Requirements

Table 1. Operating Systems and Intel® Server Boards Supported

Intel® Server Boards	Operating System Version
• Intel® Server Board S2600WT/S2600WTR family • Intel® Server Board S2600KP/S2600KPR family • Intel® Server Board S2600TP/S2600TPR family • Intel® Server Board S2600CW/S2600CWR family • Intel® Server Board S2600WF/S2600WFR family • Intel® Server Board S2600ST/S2600STR family • Intel® Server Board S2600BP/S2600BPR family • Intel® Server Board S9200WK family • Intel® Server Board D50TNP family • Intel® Server Board M50CYP family • Intel® Server Board M70KLP family	• Red Hat* Enterprise Linux* 8.1, 8.2, 7.3, 7.5 and 7.6–64 Bit. • SuSE* Linux* Enterprise Server 15, 12 Service Pack 3–64 Bit • CentOS 8.2, 7.5 and 7.3. • Ubuntu 16.04 LTS, Ubuntu 18.04 LTS and Ubuntu 20.04 LTS

1.3   Terminology

Table 2. Terminology

Term	Definition
BMC	Baseboard Management Controller
CLI	Command-Line Interface
FRU	Field Replaceable Unit
IPMI	Intelligent Platform Management Interface. Operates independently of the operating system (OS) and allows you to manage a system remotely, even in the absence of the OS.
Redfish	Operates over HTTPs using a REST API independently of the Operating System and allows you to manage a system remotely using basic https commands like post, get, patch.
LAN	Local Area Network
Management Server	Intel® Server System where SDPTool is installed. It will be acting as host server, which has network connectivity to the rest of the managed servers.
Managed Server	Intel® Server System in a cluster or data center that will be managed by Management Server.
OUT-OF-BAND	Out-of-band managed server refers to any system, which is configured with valid IPMI LAN channel and logon account to allow remote management via IPMI protocol.
SDR	Sensor Data Record
SEL	System Event Log

1.4   Related Documents

IPMI-Intelligent Platform Management Interface Specification, 2nd Generation, v2.0 (available here: http://www.intel.com/content/www/us/en/servers/ipmi/ipmi-second-gen-interface-spec-v2-rev1-1.html)

1.5   Intel® Support

Visit https://www.intel.com/content/www/us/en/support.html for current technical support information and updates.

2. Get Started

2.1 Prerequisites for Installation

The following tools must be installed prior to the installation of the Intel® SDP Tool to ensure proper functionality. The Intel® SDP Tool is an RPM based package and will fail to install if the following prerequisites are not met. These prerequisites are included with the standard distribution CD/DVD or .iso.

2.1.1 Prerequisite Packages

The following application packages are required for installing the Intel® SDP Tool:

- Python* 3.x and Python* 2
- Ipmitool* 1.8.18
- cURL* 7.29.0
- OpenSSL* 1.0.0x above
- Wget* 1.16 above
- Python*-requests
- Java* OpenJDK/Oracle* version 1.7 and above, 64-bit
- Icedtea*-web
- OpenIPMI* drivers
- defusedxml Python* module – currently not supported on Python* 2.6.9 by PIP

2.2 Installation Steps

To install the Intel® SDP Tool on the Management Server

1. Download or Copy the Intel® SDPTool Package 'SDPTool-x.y-z.tar.gz' to the target directory.

2. Untar the tar.gz file.

   ```
   Prompt #> tar -xvzmf SDPTool-x.y-z.tar.gz
   ```

3. Go to untarred 'SDPTool-x.y-z' directory folder. Run sdptool_install.sh to install the package, example below :-

   ```
   Prompt #> cd <path/to/SDPTool-x.y-z>
   Prompt #> ./sdptool_install.sh
   ```

If an older version is present, uninstall it first by using the command below:

   ```
   Prompt #> ./sdptool_uninstall.sh
   ```

Or use the update script:

   ```
   Prompt #> ./sdptool_update.sh
   ```

4. Configure the proper proxy settings using a web browser, using the below example from Mozilla* Firefox.

Note: Start the Firefox* browser > setting > advance > network settings > select auto-detect proxy settings for this network (SDP Tool's install script will install pip and then install Python* defusedxml module). Proxy may be required for getting the packages from pip. Proxy setting may also be needed to set as environment variable, contact your system/network administrator for more details. Check Section 5.8 for FAQ's.
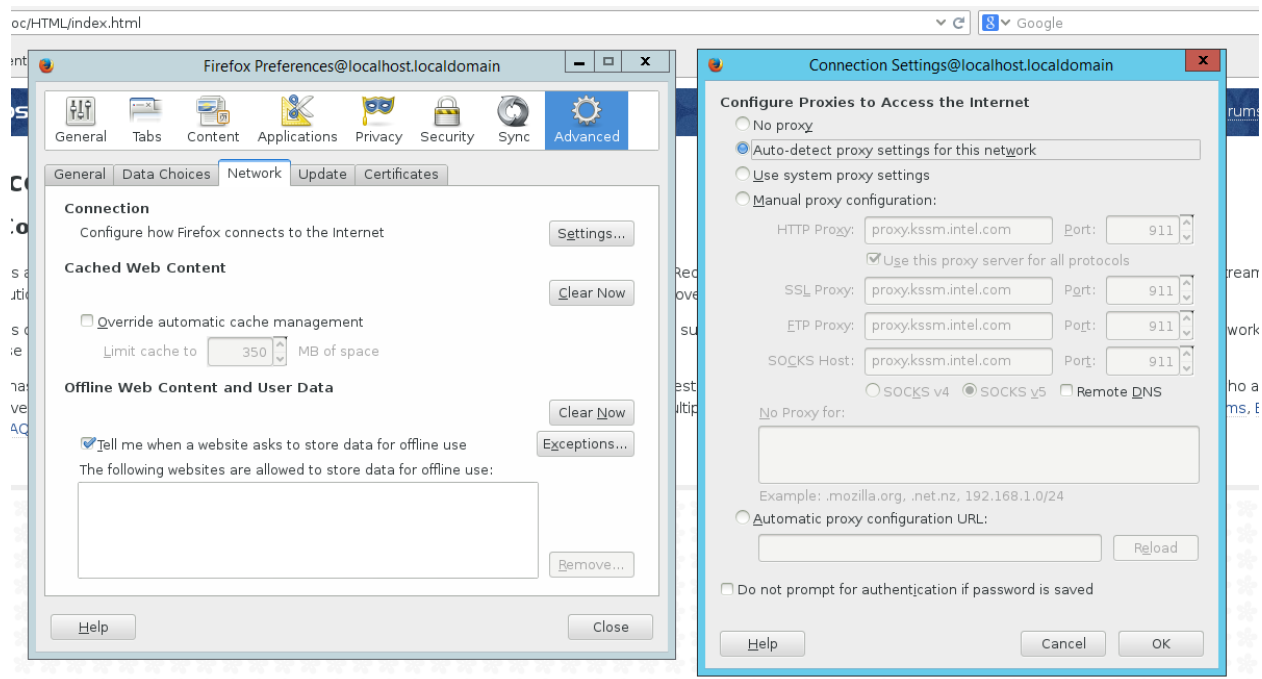
Figure 1: Proxy Settings in a Browser (Ex. Mozilla* Firefox*)

2.3 Uninstallation Steps

To uninstall the package, enter the following commands:

```
Prompt #> tar -xvzmf SDPTool-x.y-z.tar.gz

Prompt #> cd <path/to/sdptool-x.y-z>

Prompt #> ./sdptool_uninstall.sh
```

2.4 Update Steps

To update the package, enter the following commands

```
Prompt #> tar -xvzmf SDPTool-x.y-z.tar.gz

Prompt #> cd <path/to/sdptool-x.y-z>

Prompt #> ./sdptool_update.sh
```

2.5 Intel® Data Center Manager

Intel® Data Center Manager is a management data center solution stack providing accurate real-time monitoring (thermal and power), management and platform update (BIOS, BMC, etc.) capabilities on Intel® Server Boards & Systems. For more information and a feature list of the Intel® Data Center Manager, refer to the Intel® DCM user guide that comes with the download of the tool.

To support Intel® Data Center Manager, SDPTool provides the following capabilities from an XML based configuration file.

2.5.1   SDPTool Configuration File

The SDPTool configuration file is an XML file containing the following tags with no predefined (default) location for the file, passed as a command argument:

- Configuration
- SUP_Folders
- Custom_Folders
- FRU_Field

The following sample configuration file passed to the Intel® SDPTool provides the locations of various SUP and Custom folders that a user can provide instead of providing them in the command line.

```
<Configuration>
        <SUP_Folders>
                <S2600BPB>/path/to/SUP_folder</S2600BPB>
        </SUP_Folders>
        <Custom_Folders>
                <FRUField>Board Product</FRUField>
                <S2600BPB>/path/to/Custom_folder</S2600BPB>
        </Custom_Folders>
</Configuration>
```

Note: SUP is a platform update package provided by Intel. The SUP is usually distributed as a .zip file, unzipping the .zip will yield the SUP Folder. The SUP package contains the required scripts and images/binaries that will be run in the EFI shell to perform the necessary updates of the platform.

Note: The Custom Folder is a user defined folder. Users are able to write their own script that will run in the EFI shell. Refer to Section 4 for more details

2.5.2   Adding a Platform for an SUP Update

To add different platforms for an SUP based update, add the following line:

```
<Configuration>
        <SUP_Folders>
                <S2600BPB>/path/to/SUP_folder</S2600BPB>
        <S2600WFP>/path/to/SUP_folder_of_wolfpass</S2600WFP>
        </SUP_Folders>
        <Custom_Folders>
                <FRUField>Board Product</FRUField>
                <S2600BPB>/path/to/Custom_folder</S2600BPB>
        </Custom_Folders>
</Configuration>
```

Note: S2600WFP is the "Board Product" value in the FRU field of the platform.

2.5.3   Adding a Platform for a Custom Update

To add a different platform for the custom update feature, add the following line:

```
<Configuration>
      <SUP_Folders>
            <S2600BPB>/path/to/SUP_folder</S2600BPB>
      </SUP_Folders>
      <Custom_Folders>
            <FRUField>Board Product</FRUField>
            <S2600BPB>/path/to/Custom_folder</S2600BPB>
            <S2600WFP >/path/to/SUP_folder_of_wolfpass</S2600WFP >
      </Custom_Folders>
</Configuration>
```

Note: S2600WFP is the "Board Product" value in the FRU field of the platform.

1. The tags are case-sensitive.

2. The value for the FRU_Field can be obtained from one of the left columns of the fru print output. Corresponding values in the right column will form the next tag. In the above example, the FRU_Field is a "Board Product", with the value being "S2600BPB". The fru fields can be obtained by using the command in Section 3.21, with the following sample output:

```
FRU Device Description : Builtin FRU Device (ID 0)
 Chassis Type          :
 Chassis Part Number   :
 Chassis Serial        :
 Chassis Extra         :
 Chassis Extra         :
 Board Mfg Date        :
 Board Mfg             :
 Board Product         :
 Board Serial          :
 Board Part Number     :
 Product Manufacturer  :
 Product Name          :
 Product Part Number   :
 Product Version       :
 Product Serial        :
 Product Asset Tag     :
```

Figure 2: FRU Print Details

2.5.4   Sudoers

Add the DCM user to the sudoers list. SDPTool will work only as the root or if the user is part of the sudoers list.

```
#> visudo
```

Add the following line in the file:

```
<DCM_USERNAME>        ALL=(ALL)        /usr/bin/SDPTool
```

2.6   User Privileges

The IPMI interface is used for most commands. The Intel® SDPTool requires the user to have administrative privileges, otherwise the commands will return an error with a message to check the user's credentials and privileges.

2.7   Network Ports

The following network ports are used by the SDPTool to connect to the platform:

1. Ping: No TCP port is used, but ICMP packets need to be allowed.

2. https: server port outbound '**443**' and '**916'**

3. http: server port outbound '**80**'

4. ipmi: server port outbound '**623**' and '**627**'

5. kvm: server port '**5902**' and '**7578**'

6. vmedia: server port '**5123**'

Note: The Intel® SDPTool may use a combination of these ports to complete an operation.

3. Feature Script

The Intel® SDPTool script is the main engine of Intel® SDPTool OOB features. This section explains the methods to execute Intel® SDPTool features and the objectives accomplished by executing them.

3.1 General Rules

To display the usage menu, enter –h.

Example: SDPTool –h

Each valid operation run will create logs in ./Logfiles/<ip>/<operation>

Example: after running "SDPTool 192.168.1.10 bmcuser bmcpw powerstatistics"

There are log(s) in /usr/local/log/SDPTool/Logfiles/192_168_1_10/powerstatistics.log

Any error will generate error log file with .err extension. The specific definition of each error code is available in Section 6.

For reboot features, only one operation for an IP at any given time is run. Any other operation, which requires the execution of a reboot feature on the same IP will respond unexpectedly, interfering with the current operation being performed on the IP. SDPTool restricts the user from running 2 different operations that result in rebooting the same platform.

For example: SDPTool 192.168.1.10 bmcuser bmcpw getini

SDPTool 192.168.1.10 bmcuser bmcpw set_biosconfig 'QuietBoot 1'

SDPTool 192.168.1.10 bmcuser bmcpw custom_deploy customdeployfolder/

Note: These operations should not be run at the same time, as both will reboot the system and interfere with each other's operation and reboot the managed system.

Note: Some of the commands that reboot the platform require the platform to be in the KCS Policy of *"Allow All"* in order for them to work. The tool will error out if the KCS Policy is *"Restricted"* or *"Deny All".*

Note: Refer to the list of commands that perform the reboot of the platform in Section 7. Additionally, users can find the commands that would reboot the platform by running the following prompt#> SDPTool –h. This informs the user if the command will reboot the target platform.

3.2 Update Firmware

There are 2 variants of the update firmware now available via IPMI and 2 variants via Redfish:

3.2.1 SUP Folder Based

```
SDPTool <ipv4> <username> <password> update <SUP folder> [-no_user_interaction]
[-softreset]
```

To update the BIOS/ME/BMC/SDR system firmware, an SUP package must be used instead of a FSUP package. This feature makes use of flash utilities and images within the SUP package.

- -no_user_interaction: flag to reboot the system without prompt.
- -softreset: flag to soft reboot the system in case the system is in OS mode.

Example: SDPTool 192.168.1.10 admin admin123 update SUP/S2600WT

Note: SUP_Folder – Path to Update Package (SUP) is required and to be provided as argument.

3.2.2 Config File Based

```
SDPTool <ipv4> <username> <password> update –c <config file>[no_user_interaction]
[-softreset]
```

This option provides the user the ability to provide a config file instead of the SUP folder path. The creation of the config file is covered in the section 2.5. Refer to this section. The config file will be used to gather the correct SUP folder for the platform.

The other options are same as in section 3.2.1

Note: To verify the update, check the versions of the firmware. The versions can be checked using "*systeminfo*" command. Refer to section 3.23 for more details.

3.3 Custom Deploy

The custom deploy feature is update-based, using a user-defined folder to perform the action the user desires instead of using an Intel® provided SUP. Custom deploy can be used to perform various custom updates, which include, but are not limited to, firmware updates to OEM parts (SSD, NIC, HBA, etc.). Section 4 contains further information detailing the creation of a custom deployment folder and the scripts associated with the custom deploy folder.

3.3.1 Custom Folder Based

```
SDPTool <ipv4> <username> <password> custom_deploy <folder name which containing
deploy.nsh> <"argument(s) for deploy.nsh"> [-no_user_interaction] [-softreset]
```

To deploy a user customized script, the customized script must start from the deploy.nsh script.

Note: A reboot is required for this option, clearing the EFI mailbox.

- deploy_result.log: the output from deploy.nsh can be redirected to this filename; the file will be saved to Logfiles/ip folder and content will be displayed to terminal after custom_deploy script with extra argument(s) being executed
- deploy_details.log: the details from deploy.nsh can be redirected to this filename; the file will be saved to Logfiles/ip folder after custom_deploy script with extra argument(s) being executed
- -no_user_interaction: flag to reboot the system without prompt.
- -softreset: flag to soft reboot the system in case the system is in OS mode.

Example: SDPTool 192.168.1.10 admin admin123 custom_deploy folder_with_nsh_file

Example: SDPTool 192.168.1.10 admin admin123 custom_deploy folder_with_nsh_file "argument1 argument2 argument3"

3.3.2 Config File Based

```
SDPTool <ipv4> <username> <password> update –c <config file> [-
no_user_interaction] [-softreset] --all
```

This option provides the user the facility to provide a config file instead of the custom_folder path. The creation of the config file is covered in the section 2.5. Refer to this section. The config file will be used to gather the correct Custom folder for the platform.

```
--all
```

Use this option to perform a custom update. The custom_folder details are taken from the config file provided as the command argument to perform the custom update.

Note: The command used is "update" and not "custom_deploy", unlike in Section 3.3.1. The --all option is necessary to perform the custom update, otherwise the update command will perform a SUP based platform update.

3.4   Set Options

```
SDPTool <ipv4> <username> <password> setoptions <"syscfg arguments"> [-
no_user_interaction] [-softreset]
```

This option configures BIOS/BMC settings by executing syscfg command-line arguments. Refer to the syscfg user guide for specific syscfg command-line arguments.

Note: A reboot is required for this option, clearing the EFI mailbox.

- -no_user_interaction: flag to reboot the system without prompt.
- -softreset: flag to soft reboot the system in case the system is in OS mode.

Example: SDPTool 192.168.1.10 admin admin123 setoptions /i

3.5   Set Bios Config (Redfish Version of setoptions)

```
SDPTool <ipv4> <username> <password> set_biosconfig <"var new_val"> [-
no_user_interaction] [-no_reboot]
```

This option configures BIOS/BMC settings via Redfish API.

Note: A reboot is required for this option.

- -no_user_interaction: flag to reboot the system without prompt and take any input as positive nod to go ahead.
- -no_reboot: flag to make sure the operating system does not reboot. In such a case, the BIOS and ME updates cannot reflect until rebooted manually later by the user.

Example: SDPTool 192.168.1.10 admin admin123 set_biosconfig 'QuietBoot 1'

3.6   Deploy Options

```
SDPTool <ipv4> <username> <password> deployoptions <restore filename> [-
no_user_interaction] [-softreset]
```

This option configures BIOS/BMC settings by using the syscfg ini method. Once the .ini file is provided, the user may change many of the BIOS options and set them within one command.

Note: Make sure that the .ini provided for deploy options must have allowable values. A reboot is required for this option, clearing the EFI mailbox.

- -no_user_interaction: flag to reboot the system without prompt.
- -softreset: flag to soft reboot the system in case the system is in OS mode.

Example: SDPTool 192.168.1.10 admin admin123 deployoptions sysconfig.ini

3.6 Set Bios Config All (Redfish Version of deployoptions)

```
SDPTool <ipv4> <username> <password> set_biosconfig_all <restore filename> [-
no_user_interaction] [-no_reboot]
```

This option configures BIOS/BMC settings by using the syscfg ini method. Once the .ini file is provided, the user may change many of the BIOS options and set them within one command.

Note: A reboot is required for this option.

- -no_user_interaction: flag to reboot the system without prompt and take any input as positive nod to go ahead.
- -no_reboot: flag to make sure the operating system does not reboot. In such a case, the BIOS and ME updates cannot reflect until rebooted manually later by the user.

Example: SDPTool 192.168.1.10 admin admin123 set_biosconfig_all sysconfig.ini

3.7 Get Bios Options

```
SDPTool <ipv4> <username> <password> getbiosoptions <"option to retrieve"> [-
no_user_interaction] [-softreset]
```

This option returns the value of a specific bios setting that is supported by the syscfg utility. Refer to the syscfg user guide for specific syscfg command-line arguments.

Note: A reboot is required for this option, clearing the EFI mailbox.

- -no_user_interaction: flag to reboot the system without prompt.
- -softreset: flag to soft reboot the system in case the system is in OS mode.

Example: SDPTool 192.168.1.10 admin admin123 getbiosoptions "Quiet Boot"

3.8 Get Bios Config (Redfish version of getbiosoptions):

```
SDPTool <ipv4> <username> <password> get_biosconfig <"option to retrieve">
```

This option returns the value of a specific BIOS setting via Redfish API.

Note: A reboot is not required.

Example: SDPTool 192.168.1.10 admin admin123 get_biosconfig "QuietBoot"

3.9 Get INI

```
SDPTool <ipv4> <username> <password> getini [-no_user_interaction] [-softreset]
```

This command returns BIOS/BMC settings by using the syscfg /save .ini file method.

Note: A reboot is required for this option, clearing the EFI mailbox.

- -no_user_interaction: flag to reboot the system without prompt.
- -softreset: flag to soft reboot the system in case the system is in OS mode.

Example: SDPTool 192.168.1.10 admin admin123 getini

3.10 Get Bios Config all (Redfish Version of getini)

```
SDPTool <ipv4> <username> <password> get_biosconfig_all
```

This command returns BIOS/BMC settings by using Redfish API.

Note: A reboot is not required.

Example: SDPTool 192.168.1.10 admin admin123 get_biosconfig_all

3.11 KVM

```
SDPTool <ipv4> <username> <password> kvm launch
```

This command launches kvm windows for remote control.

Example: SDPTool 192.168.1.10 admin admin123 kvm launch

3.12 Vmedia

3.12.1 IPMI Based:

```
SDPTool <ipv4> <username> <password> vmedia <IMAGE/ISO> [-no_user_interaction]
```

The VMedia command allows the addition of virtual media in .img/.iso format to the remote machine.

Add the relevant virtual media by redirecting the image/iso file specified.

Note: Mounting of the same image again on the same target system is not allowed

---

Example: SDPTool 192.168.1.10 admin admin123 vmedia image.img

SDPTool 192.168.1.10 admin admin123 vmedia image.iso

3.12.2 Redfish Based:

```
SDPTool <ipv4> <username> <password> vmedia <smb://user:pass@host/file_name.ISO>
[-no_user_interaction]
```

The VMedia command allows the addition of virtual media in .iso format only to the remote machine samba share. The samba share needs to be in place beforehand for it to work.

Add the relevant virtual media by redirecting the iso file specified.

Example: SDPTool 192.168.1.10 admin admin123 vmedia smb://user:pass@host/share_dir/image.iso

3.13 IPMI

```
SDPTool <ipv4> <username> <password> ipmi <ipmitool arguments>
```

The ipmicommand is followed by arguments allowing the execution of ipmitool supported commands.

Example: SDPTool 192.168.1.10 admin admin123 ipmi lan print 3

3.14 Power

```
SDPTool <ipv4> <username> <password> power <status | on | off | cycle | reset>
```

The powercommand returns and has the ability to set the power status of a server

Example: SDPTool 192.168.1.10 admin admin123 power status

3.15 Sensor

```
SDPTool <ipv4> <username> <password> sensor
```

The sensorcommand displays the relevant sensor information of a server.

Example: SDPTool 192.168.1.10 admin admin123 sensor

3.16 SEL

```
SDPTool <ipv4> <username> <password> sel [-f <filename to save sel-log>] [-c] [-
w] [-i]
```

This command retrieves the SEL log,

Note: -i = information, -c = critical, -w = warning #-f = specifies a filename to save the SEL log.

Example: SDPTool 192.168.1.10 admin admin123 sel –w –I –f save_log.txt

3.17 Set LAN

```
SDPTool <ipv4/ipv6> <username> <password> setlan <channel> <ipv4> <mask>
<gateway> <primary dns> <secondary dns>
```

The setlan command configures the BMC LAN IP ipv4 address of a particular LAN channel.

Example: SDPTool 192.168.1.10 admin admin123 setlan 2 192.168.1.12 255.255.255.0 192.168.1.1 8.8.8.8 0.0.0.0

3.18 Disable LAN

```
SDPTool <ipv4/ipv6> <username> <password> disablelan <channel>
```

The disablelan command disables a BMC LAN IP ipv4 address of a particular LAN channel.

Example: SDPTool 192.168.1.10 admin admin123 disablelan 2

3.19 Set LAN IPV6

For the S2600WT/S2600WTR/S2600KP/S2600KPR/S2600TP/S2600TPR/S2600CW/S2600CWR/ families:

```
SDPTool <ipv4/ipv6> <username> <password> setlanipv6 <channel> <ipv6> <prefix
length[32|64|128]> <ipv6 gateway>
```

The setlanipv6 command configures the BMC LAN IP ipv6 address of a particular LAN channel.

Example: SDPTool 192.168.1.10 admin admin123 setlanipv6 2 fe80::12 64 fe80::1

For the 2600WT/S2600WTR/S2600KP/S2600KPR/S2600TP/S2600TPR/S2600CW/S2600CWR families

```
SDPTool <ipv4/ipv6> <username> <password> setlanipv6 <channel> <ipv6> <prefix
length[32|64|128]> <ipv4/6 gateway> <ipv4/6 primary dns> <ipv4/6 secondary dns>
```

The setlanipv6 command configures the BMC LAN IP ipv6 address of a particular LAN channel.

Example: SDPTool 192.168.1.10 admin admin123 setlanipv6 2 fe80::12 64 192.168.1.1 0.0.0.0 0.0.0.0

3.20 Disable LAN IPV6

```
SDPTool <ipv4/ipv6> <username> <password> disablelanipv6 <channel>
```

The diablelanipv6 command disables the BMC ipv6 LAN of a particular LAN channel.

Example: SDPTool 192.168.1.10 admin admin123 disablelanipv6 2

3.21 LAN Fail Over

```
SDPTool <ipv4> <username> <password> failover < status | enable | disable>
```

The failover command returns, sets, and disables LAN failover.

Example: SDPTool 192.168.1.10 admin admin123 failover status

3.22 Node Position

```
SDPTool <ipv4> <username> <password> nodeposition
```

The nodeposition command displays node position within a chassis, and only supports a half-width SKU.

Note: Support is available for select multi-node systems.

Example: SDPTool 192.168.1.10 admin admin123 nodeposition

3.23 System Info

```
SDPTool <ipv4> <username> <password> systeminfo
```

The systeminfo command displays the system information related to the BMC and baseboard including the BMC version, BIOS version, RMM, etc.

Example: SDPTool 192.168.1.10 admin admin123 systeminfo

3.24 FRU

```
SDPTool <ipv4> <username> <password> fru {print | set <param> <value>}
```

The frucommand displays any relevant fru information.

Example: SDPTool 192.168.1.10 admin admin123 fru print

To set fru

Example: SDPTool 192.168.1.10 admin admin123 fru set <param> <value>

3.25 Memory Info

```
SDPTool <ipv4> <username> <password> memoryinfo
```

The memoryinfo command displays any relevant memory information.

Example: SDPTool 192.168.1.10 admin admin123 memoryinfo

3.26 CPU Info

```
SDPTool <ipv4> <username> <password> cpuinfo
```

The cpuinfo command displays any relevant CPU information.

Example: SDPTool 192.168.1.10 admin admin123 cpuinfo

3.27 Memory Temperature

```
SDPTool <ipv4> <username> <password> memorytemp
```

The memorytemp command displays the temperature of the system memory.

Example: SDPTool 192.168.1.10 admin admin123 memorytemp

3.28 Power Statistics

```
SDPTool <ipv4> <username> <password> powerstatistics
```

The powerstatistic command displays system power statistics.

Example: SDPTool 192.168.1.10 admin admin123 powerstatistics

3.29 Set LAN DHCP

```
SDPTool <ipv4/ipv6> <username> <password> setlandhcp <channel>
```

The setlandhcp command sets the BMC LAN ipv4 to the dhcp of a particular LAN channel.

Example: SDPTool 192.168.1.10 admin admin123 setlandhcp 2

3.30 Set LAN DHCP IPV6

```
SDPTool <ipv4/ipv6> <username> <password> setlandhcpipv6 <channel>
```

The setlandhcpipv6 command sets the BMC LAN ipv6 to the dhcp of a particular LAN channel.

Example: SDPTool 192.168.1.10 admin admin123 setlandhcpipv6 2

3.31 Set LAN Stateless ICMPV6

```
SDPTool <ipv4/ipv6> <username> <password> setlanicmpv6 <channel>
```

The setlanicmpv6 command sets the BMC LAN ipv6 to the stateless ICMP.

Note: This operation is only supported on the following Intel® product families:
- S2600WT and S2600WTR
- S2600KP and S2600KPR
- S2600TP and S2600TPR
- S2600CW and S2600CWR
- S9200WK

Example: SDPTool 192.168.1.10 admin admin123 setlanicmpv6 2

3.32 Debug Log

```
SDPTool <ipv4> <username> <password> debuglog <filename> [-force]
```

The debuglog command fetches the BMC debug log file in .zip file format

Example: SDPTool 192.168.1.10 admin admin123 debuglog debug_log.zip

- [-force]: forces the BMC transfer mode to exit when the command is executed.

3.33 Supported Updates

```
SDPTool <ipv4> <username> <password> supportedupdates –c <config_file>
```

The supportedupdates command lists the type of updates available with the config file for the remote platform. The result will be one out of following values: None, SUP ONLY, Custom ONLY, Both

3.34 Unmount

3.34.1 IPMI Based:

```
SDPTool <ipv4> <username> <password> unmount <IMAGE/ISO>
```

The unmount command allows the user to unmount or remove media that was mounted to the remote platform using the vmedia command. The image/iso is the full path to the image that was mounted.

Note: This operation is only supported on the following Intel® product families:
- S2600WF and S2600WFR
- S2600ST and S2600STR
- S2600BP and S2600BPR
- S9200WK
- D50TNP
- M50CYP
- M70KLP

3.34.2 Redfish Based:

```
SDPTool <ipv4> <username> <password> unmount <smb://user:pass@host/file_name.ISO>
```

The unmount command allows the unmounting of virtual media in .iso format only from the remote machine samba share via Redfish. The samba share needs to be in place beforehand for it to work.

Eject the relevant virtual media by redirecting the iso file specified.

Example: SDPTool 192.168.1.10 admin admin123 unmount smb://user:pass@host/share_dir/image.iso

3.35 Storage Info

```
SDPTool <ipv4> <username> <password> storageinfo
```

The storageinfo command displays the information related to the storage devices present on the platform providing details regarding firmware version, name, etc.

Example: SDPTool 192.168.1.10 admin admin123 storageinfo

3.36 Nic Info

```
SDPTool <ipv4> <username> <password> nicinfo
```

The nicinfo command displays the information related to the networks present on the platform providing details regarding firmware version, MAC Address, etc.

Example: SDPTool 192.168.1.10 admin admin123 nicinfo

4.   custom_deploy

Custom deploy allows the user to customize or script the actions they want to perform on the platform. The script will run on the EFI shell, ensuring that the user can write scripts to perform actions based individual requirements.

When creating a custom package, the user must write a '.nsh' file and name it deploy.nsh. When executing custom_deploy as described in Section 3.3, the SDPTool will look for the deploy.nsh file within a folder that the user provides as a command-line argument .

The custom folder provided by the user is used to create an image with certain standard EFI applications and remotely mount it onto the platform. The platform then executes the deploy.nsh script to perform the necessary actions and return the results/outputs to the user. While writing the deploy.nsh file, the user can also redirect the output to 2 distinct files from which the SDPTool collects logs:

- deploy_result.log
- deploy_details.log

The deploy_result.log can be used to redirect the success or failure status of a particular action. If the user uses this redirection properly, then all success/failures will be available in this log file as *'custom_deploy_result.txt'*.

The deploy_details.log can have the detailed log of every operation in the deploy.nsh and will be available as *'custom_deploy_details.txt'.*

Note: The output of the deploy.nsh file is also captured in a separate file named *'custom_deploy_output.txt'* that will be available with the other logs in the standard location.

Note: All log files will available in the standard location that is described in Section 3.1.

In the following deploy.nsh file, actions are logged. The deploy.nsh file can be used to perform complicated operations as well and is dependent on the user's requirement.

```
echo "Starting the user custom operation" >> deploy_details.log
ls
echo "listing files : successful" >> deploy_result.log
```

The user can write deploy.nsh scripts to perform firmware updates of various components, which include, but are not limited to SDD, HBA, NIC, etc.

Note: The deploy.nsh file can be used to perform complicated operations and is dependent on the user's individual needs.
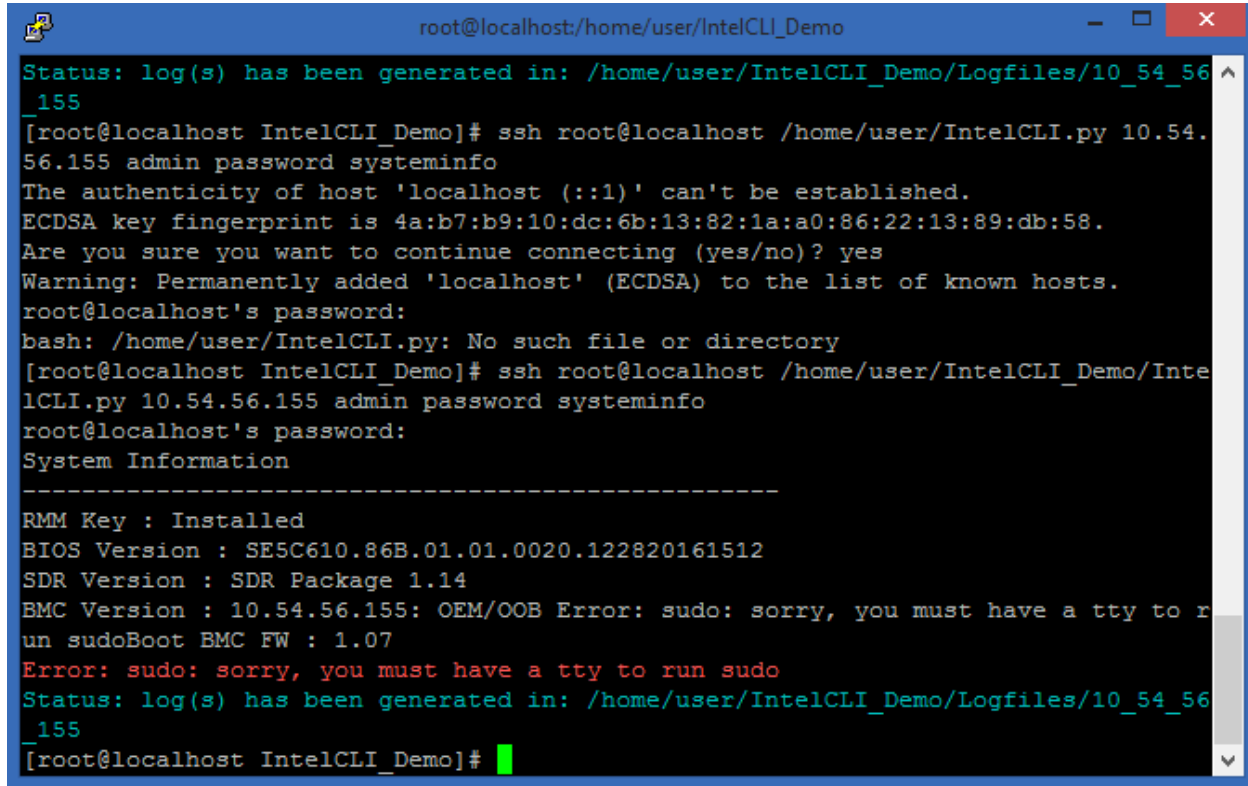
Note: Since the custom folder is provided as an argument, it will be the only folder used in the image. If any additional binary is needed when calling from the deploy.nsh file, it should be available within the directory.

5.   Troubleshooting Tips

This section lists the possible errors encountered during the use of this product with workarounds to address these errors.

5.1   SSH Command Sudo Error

Figure 3: SSH Command Error

To direct using the ssh command, add arg –t.

Example: ssh –t root@localhost SDPTool 192.168.1.10 admin admin123 powerstatistics

5.2   Tar Time Stamp Messages

Figure 4: Tar Timestamp Messages

These messages do not harm the system. To avoid seeing these messages, ensure that the date and time of the managed system are correct.

5.3 Kvm Launch glibc Version Error (SLES 11.4–64bit)

Update the glibc with the version mentioned (GLIBC_2.15). The ldd –version command can be used to check the glibc version installed on system.

Figure 5: KVM Launch

5.4 Reboot Features OOB Unable to Start on S2600WT/ S2600KP/ S2600TP/S2600CW Family (SLES11.4-64bit)

Figure 6: Reboot Features OOB

Default openssl and wget version in SLES11.4-64bit is unable to support reboot features on the S2600WT/ S2600KP/ S2600TP/ S2600CW server families. The following steps remove and upgrade openssl and wget on a SLES11.4-64bit management host.

1. Download the openssl source code 1.0.1t: https://www.openssl.org/source/old/1.0.1/

2. Remove the existing openssl:

   ```
   Prompt #> rpm -ev --nodeps openssl
   ```

3. Configure and build:

   ```
   Prompt #> tar -xvf openssl-1.0.1t.tar.gz
   Prompt #> cd openssl-1.0.1t
   Prompt #> ./config shared --prefix=/usr --openssldir=/etc/ssl --libdir=/lib
   Prompt #> make && make install
   ```

4. Download wget source code: wget http://ftp.gnu.org/gnu/wget/wget-1.15.tar.gz

5. Remove existing wget:

   ```
   Prompt #> rpm -ev --nodeps wget
   ```

6. Configure and build:

   ```
   Prompt #> tar -xvf wget-1.15.tar.gz
   Prompt #> cd wget-1.15
   Prompt #> ./configure --prefix=/usr --sysconfdir=/etc --with-ssl=openssl
   Prompt #> make && make install
   ```

5.5   Multithread Issue (RHEL 6.8-64bit)

RHEL6.8-64bit is set to 1 thread count by default since VMViewer has limited multiple simultaneous-thread support.

5.6   Soft-Reset Issue (SLES 11.4-64bit)

```
linux-b93j:/usr/local/IntelCLI # ./IntelCLI 192.168.3.54 test1 test1 getini -softreset

This operation requires the system to reset.  Proceed (y/n)?y
Status: Switching off machine: [192.168.3.54]
Error: Error in soft-reset system. Exiting.
Status: log(s) has been generated in: /usr/local/IntelCLI/Logfiles/192_168_3_54
```

Figure 7: Soft-Reset Issue

SLES 11.4-64bit will have a soft-reset issue due to the client OS prompting for a root password before shutting down the system.

Check the client system if the above error appears to ensure it is not blocked by the OS shutdown prompt.

5.7   Java* Version Required (Java* 1.7)

```
linux-37iu:/usr/local/SDPTool # ./SDPTool 192.168.11.102 test1 test1 kvm launch
Error: Error in getting java on local machine. Please make sure java is installed. Exiting.
Error: Java binary/supported version not found
    SDPTool version: 1.00.0006
    SDPTool <ipv4> <username> <password> kvm launch
```

Figure 8: Java* Version

OpenJDK/Oracle* Java* version 1.7 onwards will be required in order to run:

- kvm
- update
- customdeploy
- setoptions
- deployoptions
- getbiosoptions
- getini

To check the Java* version & provider, run:

```
Prompt #> java –version
```

5.8   Proxy Settings

Proxy settings may be required to connect to the external pip repository during install time. The defusedxml module is required for using the SDPTool's functionality that use configuration files (supportedupdates and update). update only while using -c option else defusedxml module is not required.

```
Prompt #> export http_proxy="<proxy address:proxy port>"

Prompt #> export https_proxy="<proxy address:proxy port>"
```

Set both the environment variables appropriately. Re-login may be required for the proxy to take effect.

6.   Error Codes

Table 3. Error Codes

Error Code	Error Type	Error Description
0	NoError	Success / No failure
1	ENoRMM	RMM module absent
2	ENoIPMI	IPMI module absent
3	ENoFileCreate	Error creating a file
4	ENoRetrieve	Error in retrieving the data
5	ENoProdRetrieve	Error retrieving the Product information
6	ENoSoftReset	Error trying to soft-reset
7	ENoJava	Error Java* not present
8	ENoPerm	Error not permitted/ need privileges
9	ENoConnect	Error trying to connect to the system
10	ENoRedirection	Error redirecting the image
11	ENotSupported	Error not supported currently
12	EUnsupportedPlatform	Platform not supported
13	EUnsupportedOperation	Operation not supported
14	ECurrNotSupported	Error currently not supported
15	EMissingFiles	Missing Required files
16	EMissingTags	Missing Required tag in xml
17	EMissingHardware	Missing Required Hardware
18	EMissingArgs	Missing required arguments
19	EMissingTools	Missing Required tools
20	EInterrupt	Process Interrupted
21	EInvalidArgs	Invalid Arguments
22	EInvalidIP	Invalid IP
23	EInvalidChannel	Invalid Channel
24	EInvalidSubnet	Invalid Subnet mask

Error Code	Error Type	Error Description
25	EInvalidFilename	Invalid Filename
26	EInvalidFileExt	Invalid/unexpected file extension
27	EInvalidPath	Invalid path
28	EInvalidSMBIOS	Invalid/unsupported BIOS region
29	EMismatchIPGW	IP address and Gateway are a mismatch
30	EIPMICmdError	Error running IPMI command
31	EIPMICmdTimeout	Error IPMI command timed out
32	EDupVMCLI	VMCLI already running, Duplicate Error
33	EMount	Error mounting / unmounting the image
34	EDataConvr	Error converting data
35	EKVMSessFull	Error launching KVM session is full
36	EUnknown	Unknown error
37	ESetoptionSupport	Error trying to set option
38	EOperationFail	Operation fails or reports error
39	EcurlCmd	Error from curl process
40	ESubprocess	Error invoking the process
41	ECleanupImage	Error cleaning up an Image
42	ETermDefunc	Error terminating a Defunct process
43	ETermSuspend	Error terminating a suspended process
44	EKillCmd	Error trying to kill a command
45	EStartVMCLI	Unable to start VMCLI
46	ETestapp	Testapp hits error
47	ESUPTooLarge	SUP package provided is too large
48	ESetTransMode	Error Setting transfer mode in BMC
49	ESingleFile	Error in single File
50	EFileNotFound	File / path not found
51	ESystemError	System gives error

Intel® Server Debug and Provisioning Tool User Guide

Error Code	Error Type	Error Description
52	EAbort	Aborted
53	ESysCfg	Error in using SysCfg utility
54	ERedfish	Redfish general error
55	ETimedOut	Timed out in trying to establish connection
56	ERedfishSession	Failure in creating Redfish session
57	ERedfishResponse	Redfish http response error
58	ERedfishNoSensor	Sensor not present
59	ERedfishAuthorization	Wrong username or password or hostname given
60	ERedfishUnexpected	Unexpected http Redfish error
61	ERedfishRedirect	Redirection error
62	ERedfishAttribute	Json key not present
63	EInvalidLength	Error in FRU fields length
64	ERedfishSchema	Error in Redfish schema
65	EVMRedirect	Error in virtual media redirection
66	ERecursionLimit	Limit the number of recursive calls to five at most
67	ENoStorage	No Storage device present

7. IPMI Reboot Features List

Table 4. IPMI Reboot Features List

Command	Reboot Required
cpuinfo	NO
custom_deploy	YES
debuglog	NO
deployoptions	YES
disablelan	NO
disablelanipv6	NO
failover	NO
fru	NO
getbiosoptions	YES
getini	YES
ipmi	N/A
kvm	NO
memoryinfo	NO
memorytemp	NO
nodeposition	NO
power	NO
powerstatistics	NO
sel	NO
sensor	NO
setlan	NO
setlandhcp	NO
setlandhcpipv6	NO
setlanicmpv6	NO
setlanipv6	NO
setoptions	YES
supportedupdates	NO
systeminfo	NO
unmount	NO
update	YES
vmedia	NO

8. Redfish Features List

Table 5. Redfish Features List

Command	Support	Reboot Required
cpuinfo	YES	NO
debuglog	PARTIAL*	NO
set_biosconfig_all	YES	YES
disablelan	YES	NO
disablelanipv6	NO	NO
failover	NO	NO
fru	PARTIAL**	NO
get_biosconfig	YES	NO
get_biosconfig_all	YES	NO
ipmi	NO	N/A
kvm	NO	NO
memoryinfo	YES	NO
memorytemp	YES	NO
nodeposition	NO	NO
power	YES	NO
powerstatistics	YES	NO
sel	YES	NO
sensor	YES	NO
setlan	YES	NO
setlandhcp	YES	NO
setlandhcpipv6	NO	NO
setlanicmpv6	NO	NO
setlanipv6	NO	NO
set_biosconfig	YES	YES
supportedupdates	NO	NO
systeminfo	YES	NO
unmount	YES	NO
update	YES	YES
vmedia	YES	NO
storageinfo	YES	NO
nicinfo	YES	NO

* – only the mechanism to generate the debuglog file at the target machine is supported by Redfish currently. There is no way to check if the target machine is in transfer mode from beforehand or to fetch the generated debuglog file once its generated to the local machine. It is still happening via IPMI.

** – Redfish is currently only providing option to fetch the FRU components' information and hence fru print is supported by Redfish but fru set is not and still needs IPMI.