



# **Intel® Server Debug and Provisioning Tool (Intel® SDP Tool)**

**User Guide Rev 1.3**

---

*November 2018*

# Legal Statements

---

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

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.

All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or go to: <http://www.intel.com/design/literature>.

Intel is a trademark of Intel Corporation in the U.S. and/or other countries.

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

Copyright © 2018 Intel Corporation. All rights reserved. Intel Corporation, 2200 Mission College Blvd., Santa Clara, CA 95052-8119, USA.

# Contents

---

<b>1</b>	<b>Introduction .....</b>	<b>6</b>
1.1	Document Scope.....	6
1.2	System Requirements .....	6
1.3	Terminology .....	7
1.4	Related Documents .....	7
1.5	Intel® Support .....	7
<b>2</b>	<b>Get Started .....</b>	<b>8</b>
2.1	Prerequisites for Installation.....	8
2.2	Installation Steps.....	8
2.3	Uninstallation Steps .....	9
2.4	Update Step .....	9
2.5	Intel® Data Center Manager .....	10
2.5.1	SDPTool config file.....	10
2.5.1.1	Adding a platform for SUP update .....	11
2.5.1.2	Adding a platform for Custom update .....	11
2.5.2	Sudoers .....	12
2.6	User Privileges.....	12
2.7	Network ports.....	12
<b>3</b>	<b>Feature Script.....</b>	<b>13</b>
3.1	General Rules.....	13
3.2	Update Firmware .....	14
3.2.1	SUP Folder based .....	14
3.2.2	Config File based .....	14
3.3	Custom Deploy .....	14
3.3.1	Custom Folder based .....	14
3.3.2	Config File based .....	15
3.4	Set Options .....	15
3.5	Deploy Options .....	15
3.6	Get Bios Options .....	16
3.7	Get INI .....	16
3.8	KVM .....	16
3.9	Vmedia .....	16
3.10	IPMI .....	16
3.11	Power .....	17
3.12	Sensor.....	17
3.13	SEL .....	17
3.14	Set LAN .....	17
3.15	Disable LAN.....	17
3.16	Set LAN IPV6.....	17
3.17	Disable LAN IPV6 .....	18

3.18	LAN Fail Over .....	18
3.19	Node Position .....	18
3.20	System Info .....	18
3.21	FRU .....	18
3.22	Memory Info .....	18
3.23	CPU Info .....	19
3.24	Memory Temperature .....	19
3.25	Power Statistics .....	19
3.26	Set LAN DHCP .....	19
3.27	Set LAN DHCP IPV6 .....	19
3.28	Set LAN Stateless ICMPV6 .....	19
3.29	Debug Log .....	19
3.30	Supported Updates .....	20
3.31	Unmount .....	20
<b>4</b>	<b>custom_deploy .....</b>	<b>21</b>
<b>5</b>	<b>Troubleshooting Tips .....</b>	<b>22</b>
5.1	SSH command sudo error .....	22
5.2	Tar time stamp messages .....	23
5.3	Kvm launch glibc version error (SLES 11.4-64bit) .....	23
5.4	Reboot features OOB unable to start on S2600WT/ S2600KP/ S2600TP/S2600CW family (SLES11.4-64bit) .....	23
5.5	Multithread issue (RHEL 6.8-64bit) .....	24
5.6	Soft-reset issue (SLES 11.4-64bit) .....	24
5.7	Java version required (Java 1.7) .....	25
<b>6</b>	<b>Error Codes .....</b>	<b>26</b>
<b>7</b>	<b>Reboot features list .....</b>	<b>28</b>

# Tables

---

Table 1. Operating Systems and Intel® Server Boards Supported .....	6
Table 2. Terminology .....	7
Table 3. Required tools for Intel® SDP Tool .....	8

# Figures

---

Figure 1: Proxy Setting in browser .....	9
Figure 2: FRU print details .....	12
Figure 3: SSH command error .....	22
Figure 4: Tar timestamp messages .....	23
Figure 5: KVM launch.....	23
Figure 6: Reboot features OOB .....	23
Figure 7: Soft reset issue .....	24
Figure 8: Java version.....	25

# 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 BMC Out-of-band.

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

- Intel® Server Board and System S2600WT/S2600WTR family
- Intel® Server Board S2600KP/S2600KPR family
- Intel® Server Board S2600TP/S2600TPR family
- Intel® Server Board S2600CW/S2600CWR family
- Intel® Server Board S2600WFT family
- Intel® Server Board S2600STB family
- Intel® Server Board S2600BP family

## 1.1 Document Scope

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

## 1.2 System Requirements

**Table 1. Operating Systems and Intel® Server Boards Supported**

Intel® Server Boards	Operating Systems 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 S2600WFT family Intel® Server Board S2600STB family Intel® Server Board S2600BP family	1) Red Hat Enterprise Linux* 6.8-64bit, 7.3 and 7.5 2) SuSE Linux Enterprise Server* 11 Service Pack 4-64bit and 12 Service Pack 3 3) CentOS 7.3, 7.5 and CentOS 6.8*

---

\* Refer to the Release Notes for known issues on platforms and during Install.

## 1.3 Terminology

**Table 2. Terminology**

Term	Definition
<b>BMC</b>	Baseboard Management Controller
<b>CLI</b>	Command-Line Interface
<b>FRU</b>	Field Replaceable Unit
<b>IPMI</b>	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.
<b>LAN</b>	Local Area Network
<b>Management Server</b>	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.
<b>Managed Server</b>	Intel® Server System in a cluster or data center that will be managed by Management Server.
<b>OUT-OF-BAND</b>	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.
<b>SDR</b>	Sensor Data Record
<b>SEL</b>	System Event Log

## 1.4 Related Documents

Following are the related documents for reference:

IPMI-Intelligent Platform Management Interface Specification, 2<sup>nd</sup> Generation, v2.0 (available [here](#))

## 1.5 Intel® Support

Visit [http://www.intel.com/p/en\\_US/support/](http://www.intel.com/p/en_US/support/) to get the latest and most complete technical support information.

For an updated support contact list, see <http://www.intel.com/support/9089.htm/>

## 2 Get Started

---

### 2.1 Prerequisites for Installation

The following is the list of tools that are required for the functioning of the Intel® SDP Tool, These have to be installed prior to the installation of the Intel® SDP Tool. The Intel® SDP Tool is a RPM based package and will fail to install if the following prerequisites are not already installed. The prerequisites required are part of the standard distribution CD/DVD or .iso.

**Table 3. Required tools for Intel® SDP Tool**

Required Packages
<b>Python 2.7.5, Python 2.6.9</b>
<b>Ipmitool 1.8.13</b>
<b>curl 7.29.0</b>
<b>Openssl 1.0.0x above</b>
<b>Wget 1.16 above</b>
<b>Python-requests</b>
<b>Java OpenJDK/Oracle version 1.7 and above 64bit</b>
<b>Icedtea-web</b>
<b>OpenIPMI drivers</b>

### 2.2 Installation Steps

To install Intel® SDP Tool on the Management Server

1. Download or Copy the Intel® SDPTool Package 'SDPTool-x.y-z.tar.gz' to 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. Use the command below:-

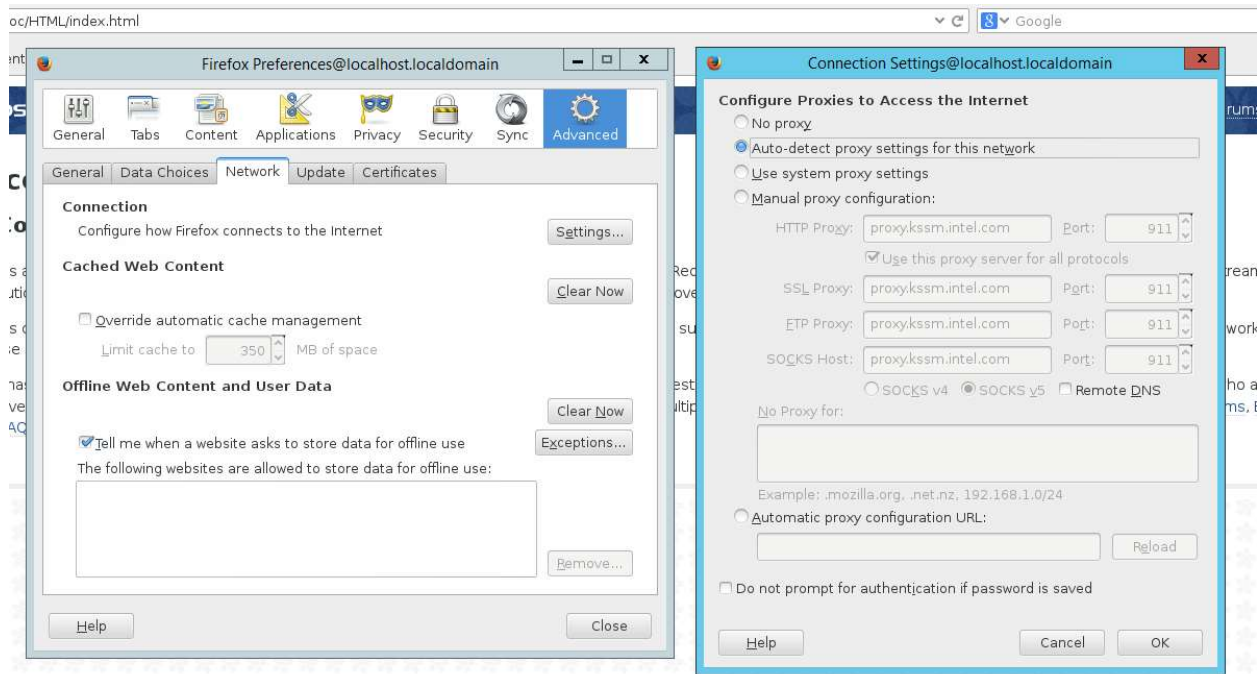
Prompt #> ./sdptool\_uninstall.sh

Or use update script

Prompt #> ./sdptool\_update.sh

4. Go to your Web browser and configure proper proxy settings, below is an example for Firefox Browser :-

- a. Go to Firefox browser > setting > advance > network settings > select auto-detect proxy settings for this network (\*This step is only required for launching KVM)



**Figure 1: Proxy Setting in browser**

## 2.3 Uninstallation Steps

To Uninstall the package use the following command:-

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

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

Prompt #> ./sdptool\_uninstall.sh

## 2.4 Update Step

To update the package use 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 info and features list of Intel® Data Center Manager please 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, the configuration of the capabilities is based on a XML based config file.

### 2.5.1 SDPTool config file

SDPTool config file is a XML file having the following tags, there is no pre-defined (default) location for the file which is passed as command argument:

1. Configuration
2. SUP\_Folders
3. Custom\_Folders
4. FRU\_Field

Below are the contents of the sample config file passed to Intel® SDPTool to provide the locations of the various SUP folder<sup>†</sup> and Custom folders<sup>‡</sup> that 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>
```

In the above example we are providing the path to the SUP folder and custom deploy folder within the XML tags <SUP\_Folders> and <Custom\_Folders> respectively.

---

<sup>†</sup> SUP is a platform update package provided by Intel® for updating the platform. 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.

<sup>‡</sup> Custom Folder is a user defined folder. Users can write their own script that will run in the efi shell. Please refer to section 4 for the details of creating a Custom Folder.

### 2.5.1.1 Adding a platform for SUP update

To add different platform for SUP based update add the following line as below:-

```
<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.

### 2.5.1.2 Adding a platform for Custom update

To add a different platform for custom update add the line as shown below:-

```
<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 next tag. In the above example FRU\_Field is “Board Product”, the values for which is “S2600BPB”, this will vary for different platforms. The fru fields can be obtained by using the command in section 3.21, the sample output is shown in the figure below named FRU print details:

```

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**

## 2.5.2 Sudoers

Add the DCM user to the sudoers list. SDPTool will work only as 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

For most of the commands we use the IPMI interface. The user credentials used with the SDPTool require user to be of “admin” privilege level else the commands will error out with the message to check the user credentials and privilege.

## 2.7 Network ports

The following are the network ports that will be used by the SDPTool for connecting to the platform:

1. Ping – No TCP port is used, but ICMP packets needs to be allowed.
2. https – server port outbound ‘**443**’
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: SDPTool may use a combination of these ports to complete an operation.

## 3 Feature Script

---

Intel® SDPTool script is the main engine of Intel® SDPTool OOB features. This section explains the ways to execute Intel® SDPTool features, and the objectives the user can accomplish by using them.

### 3.1 General Rules

1. To display usage menu

Example: SDPTool -h

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

Example: after running “SDPTool 192.168.1.10 bmcuser bmcpw powerstatistics”

There will be log(s) in /usr/local/log/SDPTool/Logfiles/192\_168\_1\_10/powerstatistics.log

3. Any failure will generate a \*.err log file. Meaning of error code can be found in section 6.

4. For **reboot features**<sup>§</sup>, it is restricted to run one such operation for an IP at any given time. Any other operation which require reboot features execution on the same IP will have unexpected behavior since it will interfere with current operation being performed on the IP. SDPTool restricts user from running 2 different operations that result in rebooting the same platform.

For ex: SDPTool 192.168.1.10 bmcuser bmcpw getini

SDPTool 192.168.1.10 bmcuser bmcpw custom\_deploy  
customdeployfolder/

5. The above operations should not be run at the same time since both will reboot the system and interfere with each other's operation, also rebooting the managed system.

---

<sup>§</sup> Reboot features - The SDPTool commands that will reboot the platform for performing the necessary actions. The list of commands that will perform the reboot of the platform is listed in section 7, additionally user can find the commands that would reboot the platform by running the following

*prompt#> SDPTool -h*

The help will let the user know if the command will reboot the target platform.

## 3.2 Update Firmware

There are 2 different variants of the update firmware now available:

### 3.2.1 SUP Folder based

**SDPTool <ipv4> <username> <password> update <SUP folder> [-no\_user\_interaction] [-softreset]**

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

Example: SDPTool 192.168.1.10 admin admin123 update SUP/S2600WT

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 facility to provide a config file instead of the SUP folder path. The creation of the config file is covered in the section 2.5. Please 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

## 3.3 Custom Deploy

The custom deploy feature is based on update. The custom deploy uses a user defined folder to perform the action user desires instead of using Intel® provided SUP. The custom deploy can be used to perform various custom updates which include but not limited to firmware updates to OEM parts such as SSD, NIC, HBA, etc. The section 4 has the complete details of how to create a custom deploy 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 user customized script, the customized script must start from deploy.nsh script. (\*Require reboot and EFI mailbox will be cleared)

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

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. Please 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 must to perform the custom update else 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]**

To configure BIOS/BMC settings by executing syscfg command line arguments (\*requires reboot and EFI mailbox will be cleared). Refer to syscfg user guide for the syscfg command line arguments.

-no\_user\_interaction: flag to reboot the system without prompt; -softreset: flag to soft reboot the system in case the system is in OS

Example: SDPTool 192.168.1.10 admin admin123 setoptions /i

## 3.5 Deploy Options

**SDPTool <ipv4> <username> <password> deployoptions <restore filename> [-no\_user\_interaction] [-softreset]**

To configure BIOS/BMC settings by using syscfg ini method. (\*requires reboot and EFI mailbox will be cleared). The ini file is to be provided, User can change many of the BIOS options and set them in this one command.

-no\_user\_interaction: flag to reboot the system without prompt; -softreset: flag to soft reboot the system in case the system is in OS

Example: SDPTool 192.168.1.10 admin admin123 deployoptions sysconfig.ini

## 3.6 Get Bios Options

**SDPTool <ipv4> <username> <password> getbiosoptions <"option to retrieve"> [-no\_user\_interaction] [-softreset]**

To get the value of a particular bios settings which is supported by syscfg utility (\*requires reboot and EFI mailbox will be cleared). Refer to syscfg user guide for the syscfg command line arguments.

-no\_user\_interaction: flag to reboot the system without prompt; -softreset: flag to soft reboot the system in case the system is in OS

Example: SDPTool 192.168.1.10 admin admin123 getbiosoptions "Quiet Boot"

## 3.7 Get INI

**SDPTool <ipv4> <username> <password> getini [-no\_user\_interaction] [-softreset]**

To get BIOS/BMC settings by using syscfg /save .ini file method (\*requires reboot and EFI mailbox will be cleared)

-no\_user\_interaction: flag to reboot the system without prompt; -softreset: flag to soft reboot the system in case the system is in OS

Example: SDPTool 192.168.1.10 admin admin123 getini

## 3.8 KVM

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

To launch kvm windows for remote control

Example: SDPTool 192.168.1.10 admin admin123 kvm launch

## 3.9 Vmedia

**SDPTool <ipv4> <username> <password> vmedia <IMAGE/ISO> [-no\_user\_interaction]**

VMedia allows to add a Virtual Media in .img/.iso format to the remote machine.

Add virtual media by redirecting image/iso file specified

Example: SDPTool 192.168.1.10 admin admin123 vmedia image.img

SDPTool 192.168.1.10 admin admin123 vmedia image.iso

## 3.10 IPMI

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

ipmi command followed by arguments allows to execute ipmitool supported commands

Example: SDPTool 192.168.1.10 admin admin123 ipmi lan print 3



## 3.11 Power

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

To get/set power status of a server

Example: SDPTool 192.168.1.10 admin admin123 power status

## 3.12 Sensor

**SDPTool <ipv4> <username> <password> sensor**

To display sensors information of a server

Example: SDPTool 192.168.1.10 admin admin123 sensor

## 3.13 SEL

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

To retrieve SEL log (\* -i = information, -c = critical, -w = warning # -f = specific a filename to save the SEL log)

Example: SDPTool 192.168.1.10 admin admin123 sel -w -I -f save\_log.txt

## 3.14 Set LAN

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

To configure 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

## 3.15 Disable LAN

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

To disable a BMC LAN IP ipv4 address of a particular LAN channel

Example: SDPTool 192.168.1.10 admin admin123 disablelan 2

## 3.16 Set LAN IPV6

*For S2600WT/S2600WTR/S2600KP/S2600KPR/S2600TP/S2600TPR/S2600CW/S2600CWR family*

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

To configure 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 S2600WF/S2600WFR/S2600SW/S2600SWR/S2600BP/S2600BPR family*

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

To configure 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

## 3.17 Disable LAN IPV6

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

To disable BMC ipv6 LAN of a particular LAN channel

Example: SDPTool 192.168.1.10 admin admin123 disablelanipv6 2

## 3.18 LAN Fail Over

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

To get/set/disable LAN fail over

Example: SDPTool 192.168.1.10 admin admin123 failover status

## 3.19 Node Position

**SDPTool <ipv4> <username> <password> nodeposition**

To display node position in a chassis. Only supports half-width SKU (\*support available for select multi-node systems only)

Example: SDPTool 192.168.1.10 admin admin123 nodeposition

## 3.20 System Info

**SDPTool <ipv4> <username> <password> systeminfo**

Displays the system information related to BMC and baseboard, which includes BMC version, BIOS version, RMM, etc.

Example: SDPTool 192.168.1.10 admin admin123 systeminfo

## 3.21 FRU

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

To display 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>

## 3.22 Memory Info

**SDPTool <ipv4> <username> <password> memoryinfo**

To display memory info

Example: SDPTool 192.168.1.10 admin admin123 memoryinfo

## 3.23 CPU Info

**SDPTool <ipv4> <username> <password> cpuinfo**

To display CPU info

Example: SDPTool 192.168.1.10 admin admin123 cpuinfo

## 3.24 Memory Temperature

**SDPTool <ipv4> <username> <password> memorytemp**

To display temperature of system memory

Example: SDPTool 192.168.1.10 admin admin123 memorytemp

## 3.25 Power Statistics

**SDPTool <ipv4> <username> <password> powerstatistics**

To display system power statistics

Example: SDPTool 192.168.1.10 admin admin123 powerstatistics

## 3.26 Set LAN DHCP

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

To set BMC LAN ipv4 to dhcp of a particular LAN channel

Example: SDPTool 192.168.1.10 admin admin123 setlandhcp 2

## 3.27 Set LAN DHCP IPV6

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

To set BMC LAN ipv6 to dhcp of a particular LAN channel

Example: SDPTool 192.168.1.10 admin admin123 setlandhcpipv6 2

## 3.28 Set LAN Stateless ICMPV6

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

To set BMC LAN ipv6 to stateless ICMP (\*This operation is supported on [S2600WT/S2600WTR/S2600KP/S2600KPR/S2600TP/S2600TPR/S2600CW/S2600CWR family](#) only)

Example: SDPTool 192.168.1.10 admin admin123 setlanicmpv6 2

## 3.29 Debug Log

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

Fetches the BMC debug log in zip file format

Example: SDPTool 192.168.1.10 admin admin123 debuglog debug\_log.zip

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

## 3.30 Supported Updates

**SDPTool <ipv4> <username> <password> supportedupdates -c <config\_file>**

To list the type of updates that are available with the config file for the remote platform. The result will be one out of following values: None, SUP ONLY, Custom ONLY, Both

## 3.31 Unmount

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

The command allows 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. Currently only S2600WF/S2600WFR/S2600SW/S2600SWR/S2600BP/S2600BPR family of platforms are supported.

## 4 custom\_deploy

---

Custom deploy allows user to customize/script the actions they want to perform on the platform. The script will run on the efi shell so the user can write scripts to perform actions based on his/her requirement.

For creating a custom package user needs to write an '.nsh' file and will name it as '**deploy.nsh**'. The SDPTool when trying to perform custom\_deploy as mentioned in section 3.3 will look for a folder that the user provides as command-line-argument, in this folder it will look for the '**deploy.nsh**' file.

The custom folder so provided will be used to create an image with certain standard efi applications and remotely mount it onto the platform. The platform will then execute the deploy.nsh script to perform the necessary actions and get the results/outputs back to the user. The user while writing the deploy.nsh can redirect the output to 2 distinct files from which SDPTool collects the logs:

1. deploy\_result.log
2. deploy\_details.log

The deploy\_result.log can be used to redirect the success or failure status of a particular action. If user uses this redirection properly then all of the 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 is also captured in a separate file called the '**custom\_deploy\_output.txt**' that will be available with the other logs in the standard location.

All the log files will be available in the standard location as mentioned in section 3.1. Here's a sample deploy.nsh file

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

In the above deploy.nsh we are logging our actions, the deploy.nsh can be used to perform complicated operations as well and is dependent on the user's requirement.

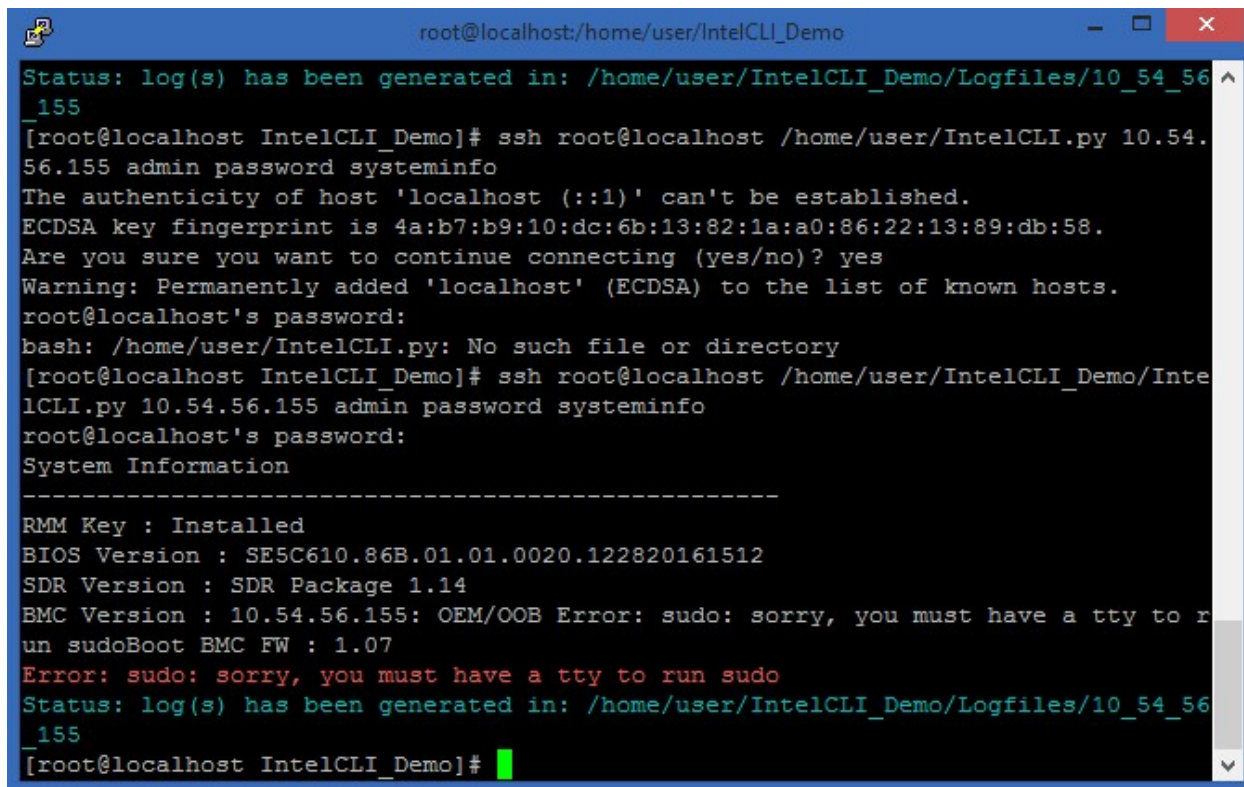
The user can write such deploy.nsh scripts to perform firmware updates of various components which include SDD, HBA, NIC, etc. but are not limited to perform only these operations, user can script much more complicated actions.

Note: since the custom folder is provided as argument only this folder will be used in the image, so if user wants to add any of his binary that he/she would like to call from deploy.nsh they should be available within the directory.

## 5 Troubleshooting Tips

This section lists the possible errors you may encounter during the use of this product, and workarounds to address the errors.

### 5.1 SSH command sudo error

A screenshot of a terminal window titled 'root@localhost:/home/user/IntelCLI\_Demo'. The terminal shows the following sequence of events: 1. A status message: 'Status: log(s) has been generated in: /home/user/IntelCLI\_Demo/Logfiles/10\_54\_56\_155'. 2. An SSH command: '[root@localhost IntelCLI\_Demo]# ssh root@localhost /home/user/IntelCLI.py 10.54.56.155 admin password systeminfo'. 3. SSH connection messages: 'The authenticity of host 'localhost (::1)' can't be established. ECDSA key fingerprint is 4a:b7:b9:10:dc:6b:13:82:1a:a0:86:22:13:89:db:58. Are you sure you want to continue connecting (yes/no)? yes' and 'Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.' 4. Password prompt: 'root@localhost's password:'. 5. Bash error: 'bash: /home/user/IntelCLI.py: No such file or directory'. 6. Another SSH command: '[root@localhost IntelCLI\_Demo]# ssh root@localhost /home/user/IntelCLI\_Demo/IntelCLI.py 10.54.56.155 admin password systeminfo'. 7. Password prompt: 'root@localhost's password:'. 8. System information output: 'System Information', followed by a separator line and details: 'RMM Key : Installed', 'BIOS Version : SE5C610.86B.01.01.0020.122820161512', 'SDR Version : SDR Package 1.14', 'BMC Version : 10.54.56.155: OEM/OOB Error: sudo: sorry, you must have a tty to run sudoBoot BMC FW : 1.07'. 9. A red error message: 'Error: sudo: sorry, you must have a tty to run sudo'. 10. Another status message: 'Status: log(s) has been generated in: /home/user/IntelCLI\_Demo/Logfiles/10\_54\_56\_155'. 11. The prompt returns to '[root@localhost IntelCLI\_Demo]#'.

```
root@localhost:/home/user/IntelCLI_Demo
Status: log(s) has been generated in: /home/user/IntelCLI_Demo/Logfiles/10_54_56_155
[root@localhost IntelCLI_Demo]# ssh root@localhost /home/user/IntelCLI.py 10.54.56.155 admin password systeminfo
The authenticity of host 'localhost (::1)' can't be established.
ECDSA key fingerprint is 4a:b7:b9:10:dc:6b:13:82:1a:a0:86:22:13:89:db:58.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
root@localhost's password:
bash: /home/user/IntelCLI.py: No such file or directory
[root@localhost IntelCLI_Demo]# ssh root@localhost /home/user/IntelCLI_Demo/IntelCLI.py 10.54.56.155 admin password systeminfo
root@localhost's password:
System Information
-----
RMM Key : Installed
BIOS Version : SE5C610.86B.01.01.0020.122820161512
SDR Version : SDR Package 1.14
BMC Version : 10.54.56.155: OEM/OOB Error: sudo: sorry, you must have a tty to run sudoBoot BMC FW : 1.07
Error: sudo: sorry, you must have a tty to run sudo
Status: log(s) has been generated in: /home/user/IntelCLI_Demo/Logfiles/10_54_56_155
[root@localhost IntelCLI_Demo]#
```

Figure 3: SSH command error

To direct using ssh command, you need to add arg -t

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



## 5.2 Tar time stamp messages

```
tar: cmdtoolX64.efi: time stamp 2017-04-12 22:37:02 is 115999883.204964241 s in the future
tar: efifmt.efi: time stamp 2017-04-12 22:37:22 is 115999903.20435322 s in the future
tar: ipmi.efi: time stamp 2017-04-13 17:23:02 is 116067443.203744765 s in the future
tar: mkedk2ramdiskX64.efi: time stamp 2017-04-12 22:37:42 is 115999923.20330491 s in the future
tar: mkramdiskX64.efi: time stamp 2015-02-05 15:55:54 is 47115015.202896016 s in the future
tar: RamDiskDxe.efi: time stamp 2017-04-14 02:56:44 is 116101865.202739364 s in the future
tar: ramdisk.efi: time stamp 2015-02-05 15:55:54 is 47115015.202524166 s in the future
tar: rundeploy.nsh: time stamp 2017-04-18 17:27:00 is 116499681.202410696 s in the future
tar: Startup.nsh: time stamp 2017-04-18 17:22:02 is 116499383.202309794 s in the future
tar: syscfg.efi: time stamp 2017-02-27 21:05:30 is 112192791.201309007 s in the future
tar: syscfg_temp.efi: time stamp 2017-02-27 21:05:32 is 112192793.18213325 s in the future
tar: vmdrive_map: time stamp 2017-03-16 21:53:50 is 113664491.18204245 s in the future
```

**Figure 4: Tar timestamp messages**

These messages do no harm. To avoid seeing these messages make sure the date-time of managed system is correct.

## 5.3 Kvm launch glibc version error (SLES 11.4-64bit)



**Figure 5: KVM launch**

Update the glibc with the version mentioned (GLIBC\_2.15). `ldd -version` command can be used to check the glibc version installed on system.

## 5.4 Reboot features OOB unable to start on S2600WT/ S2600KP/ S2600TP/S2600CW family (SLES11.4-64bit)

```
linux-37iu:/usr/local/SDPTool # ./SDPTool 192.168.11.164 test1 test1 getini
Status: Hard-reset system by default.

This operation requires the system to reset. Proceed (y/n)?y
Status: iso/mountc_192.168.11.164 exists
Status: Starting VMCLI...
Error: Redirection is not started. Exiting.
Status: log(s) has been generated in: /usr/local/SDPTool/Logfiles/192_168_11_164
```

**Figure 6: Reboot features OOB**

1. Default openssl and wget version in SLES11.4-64bit is unable to support reboot features on S2600WT/ S2600KP/ S2600TP/ S2600CW family platform. Please follow the steps to remove and upgrade openssl and wget on

SLES11.4-64bit management host.

2. Download openssl source code 1.0.1t -  
<https://www.openssl.org/source/old/1.0.1/>
3. Remove existing openssl:  
`rpm -ev --nodeps openssl`
4. Configure and build:  
`tar -xvf openssl-1.0.1t.tar.gz`  
`cd openssl-1.0.1t`  
`./config shared --prefix=/usr --openssldir=/etc/ssl --libdir=/lib`  
`make && make install`
5. Download wget source code - wget <http://ftp.gnu.org/gnu/wget/wget-1.15.tar.gz>
6. Remove existing wget:  
`rpm -ev --nodeps wget`
7. Configure and build:  
`tar -xvf wget-1.15.tar.gz`  
`cd wget-1.15`  
`./configure --prefix=/usr --sysconfdir=/etc --with-ssl=openssl`  
`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 support for supporting multiple thread starting at same time

## 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**

1. SLES 11.4-64bit will have soft-reset issue since the client OS will prompt for root password before shutting down system
2. Please check the client system if the above error appears to make sure the client system is not blocked by 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**

1. OpenJDK/Oracle Java version 1.7 onwards will be required in order to run kvm, update, customdeploy, setoptions, deployoptions, getbiosoptions, and getini
2. To check java version & provider, run:  
java -version

## 6 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 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
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	EKVMSSessFull	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
52	EAbort	Aborted

## 7 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