

Lenovo™

# Lenovo ThinkServer System Manager Command Line Interface User Guide



ThinkThinkThinkServerThink

TSMCLI

**Second Edition (May 2018)**  
**© Copyright Lenovo 2014-2018.**

LIMITED AND RESTRICTED RIGHTS NOTICE: If data or software is delivered pursuant a General Services Administration "GSA" contract, use, reproduction, or disclosure is subject to restrictions set forth in Contract No. GS-35F-05925.

## Table of Contents

Table of Contents.....	3
1. Introduction.....	6
2. System Requirements.....	7
3. Install Procedure.....	8
4. Uninstall Procedure.....	12
5. Usage.....	13
6. Commands Output View.....	14
7. Commands Reference.....	16
backup-config.....	17
clear-sel.....	18
create-adrolegroup.....	19
create-alertnotification.....	20
create-alertpolicy.....	22
create-credential.....	24
create-eventfilter.....	26
create-firewallprule.....	32
create-firewallportrule.....	34
create-landestination.....	36
create-ldaprolegroup.....	38
create-user.....	40
delete-adrolegroup.....	42
delete-alertpolicy.....	43
delete-eventfilter.....	44
delete-firewallprule.....	45
delete-firewallportrule.....	46
delete-landestination.....	47
delete-ldaprolegroup.....	48
delete-user.....	51
enter-fwupdatemode.....	52
exit-fwupdatemode.....	53
firmware-update.....	54
get-adrolegroups.....	57
get-adsettings.....	59
get-alertpolicies.....	60
get-auditlog.....	62
get-auditlogsettings.....	63
get-commands.....	64
get-crashscreen.....	65
get-datetime.....	66
get-dns.....	67
get-eventfilters.....	68
get-firewallrules.....	70
get-frus.....	73
get-fwupdatestatus.....	75
get-fwversions.....	78
get-help.....	80
get-identifierstate.....	81

get-landestinations.....	82
get-ldaprolegroups.....	83
get-ldapsettings.....	85
get-niclinksettings.....	87
get-nics.....	88
get-nicsettings.....	89
get-ntpsettings.....	91
get-pamorder.....	92
get-powerbuttonstate.....	93
get-powerconsumption.....	94
get-powerconsumptionsettings.....	96
get-powerstate.....	97
get-psuredundancysettings.....	98
get-remotemediasettings.....	100
get-sel.....	102
get-sensors.....	103
get-sensortypes.....	105
get-services.....	106
get-smtpsettings.....	107
get-solsettings.....	109
get-sslcertificate.....	110
get-systeminfo.....	113
get-timezones.....	114
get-users.....	115
get-websessiontimeout.....	117
reset-config.....	118
reset-powerconsumptionhistory.....	119
restore-config.....	120
set-adrolegroup.....	121
set-adsettings.....	123
set-alertpolicy.....	125
set-auditlogsettings.....	127
set-datetime.....	128
set-dns.....	130
set-eventfilter.....	132
set-identifierstate.....	137
set-landestination.....	138
set-ldaprolegroup.....	140
set-ldapsettings.....	142
set-niclinksettings.....	145
set-nicsettings.....	147
set-ntpsettings.....	150
set-pamorder.....	152
set-powerbuttonstate.....	153
set-powerconsumptionsettings.....	154
set-powerstate.....	155
set-psuredundancysettings.....	156
set-remotemediasettings.....	158
set-service.....	160
set-smtpsettings.....	162
set-solsettings.....	165

set-sslcertificate.....	167
set-timezone.....	168
set-user.....	169
set-websessiontimeout.....	171
start-fwupdate.....	172
start-remotemediainage.....	174
stop-pendingupdates.....	175
stop-remotemediainage.....	176
transfer-fwimage.....	177
upload-fwimage.....	179
8. Commands permissions.....	180
9. Ports Usage.....	183
10. ThinkServer firmware update flow.....	184
11. Exceptions and Errors.....	186
12. Tips.....	187
12.1. How to enable alert notifications for the TSM using TSMCLI.....	187
12.2. Unsupported special characters on the TSM and TSMCLI.....	189

## 1. Introduction

This guide will provide instruction about how to use the features available in the **Lenovo ThinkServer System Manager Command Line Interface (TSMCLI)**.

The Lenovo ThinkServer System Manager (TSM) is a Baseboard Management Controller (BMC). A BMC is an embedded hardware component present in servers to monitor and control their status.

The TSMCLI is a Microsoft® PowerShell module that provides a set of tools for monitoring and controlling the TSM of your ThinkServer. With the help of TSMCLI administrators can manage users, network settings, platform event filters and other TSM functionalities of one or more ThinkServer units in a scriptable way. It also provides some tools for managing ThinkServer operations like power control and remote media images.

To make use of TSMCLI, basic knowledge of Microsoft® PowerShell is required. More information about Microsoft® PowerShell can be found on the following links:

[Getting Started with Windows PowerShell](#) and [Windows PowerShell User's Guide](#).

The next sections will present all commands and options available in the TSMCLI.

## 2. System Requirements

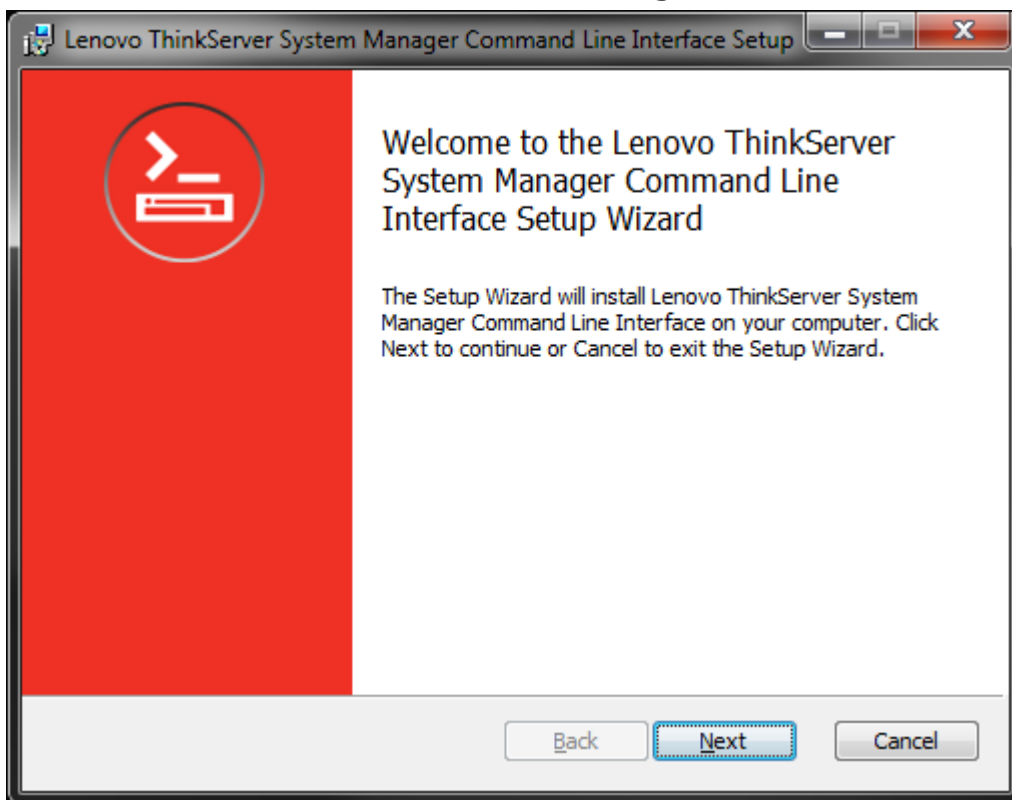
In order to execute the TSMCLI your system must meet the following requirements:

- Lenovo ThinkServer RD350, RD450, RD550, RD650 or TD350;
- Microsoft® Windows 7, 8, 8.1, Server 2008 R2, 2012, 2016 or version 1709 32-bit or 64-bit operating system;
- Microsoft® PowerShell 2, 3, 4 or 5 installed;
- Microsoft® .NET framework 3.5.

### 3. Install Procedure

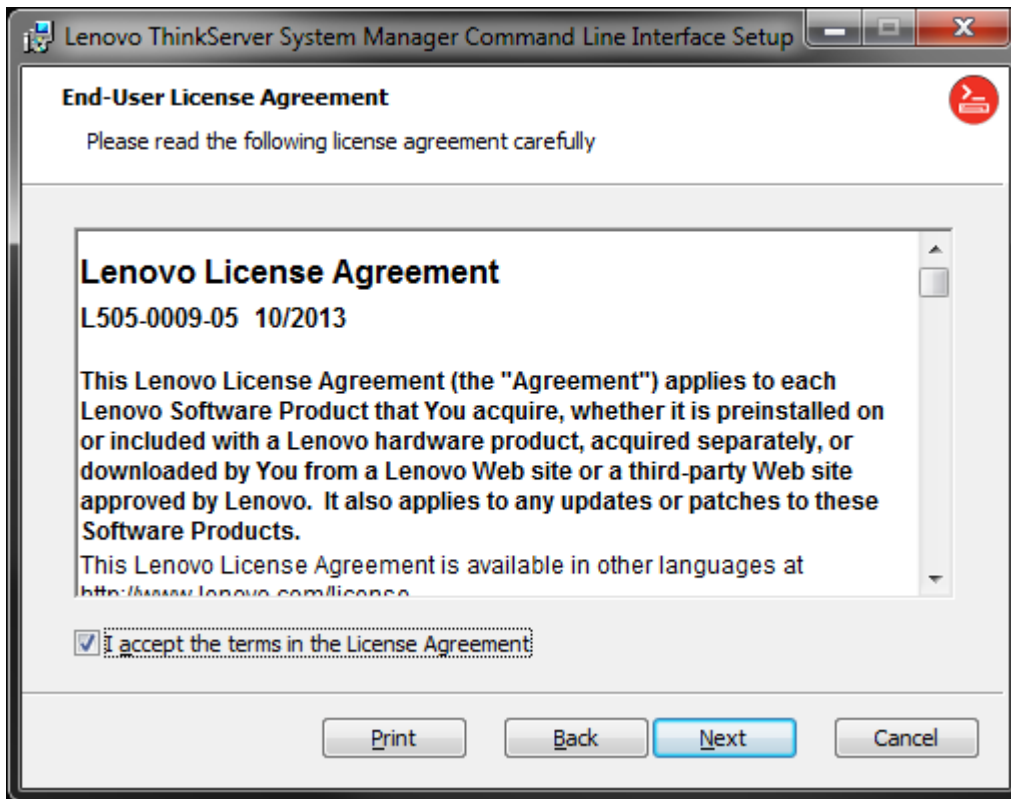
In order to install the TSMCLI follow the instructions below:

1. Download the latest TSMCLI installer.
  - To download TSMCLI, go to [http://support.lenovo.com/en\\_US/downloads/default.page](http://support.lenovo.com/en_US/downloads/default.page) and follow the instructions on the Web page.
2. Execute the installer with **Administrator Privileges**.

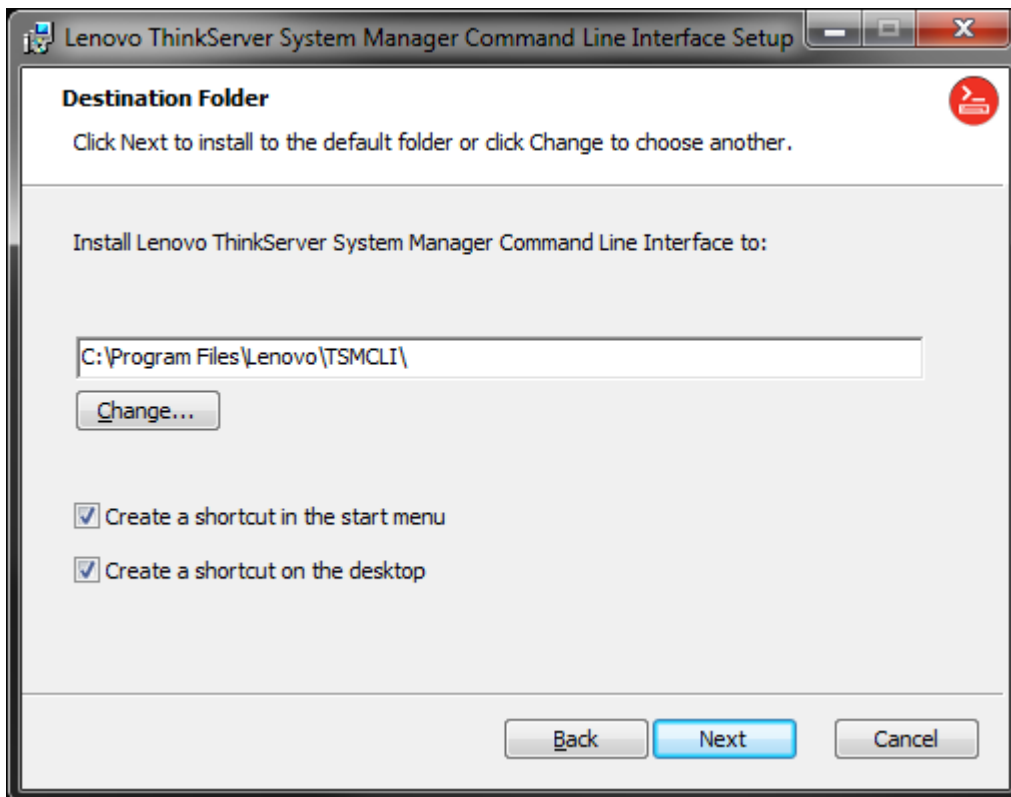




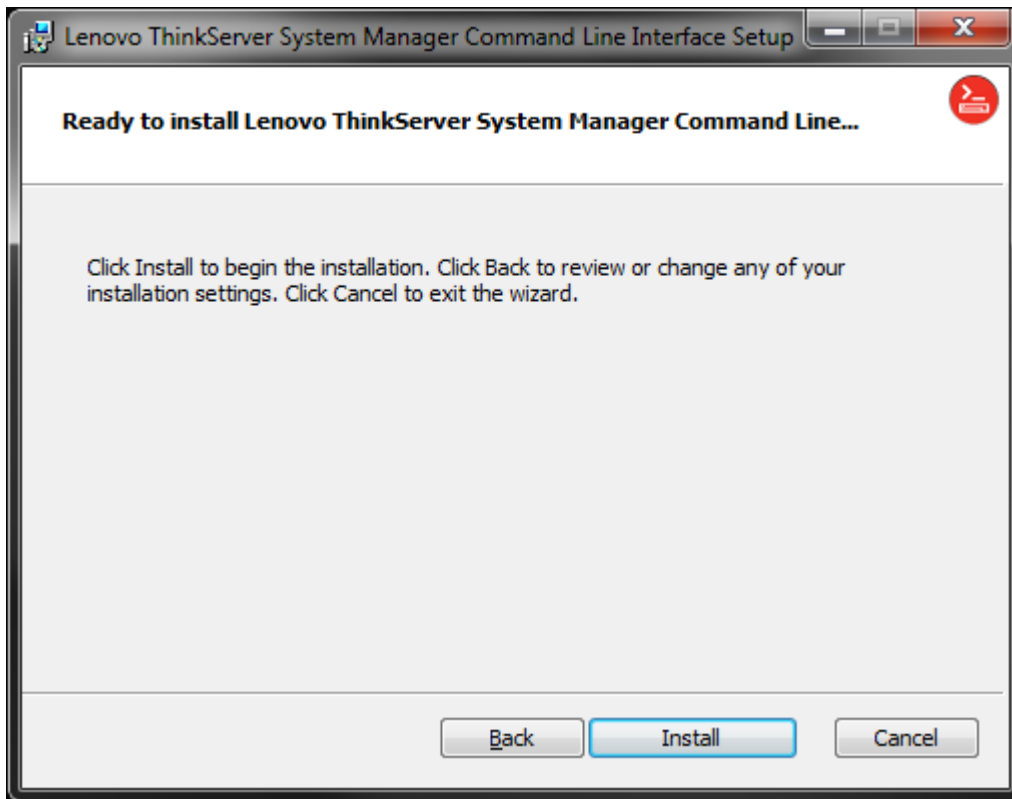
## 3. Read and accept the EULA for the TSMCLI.



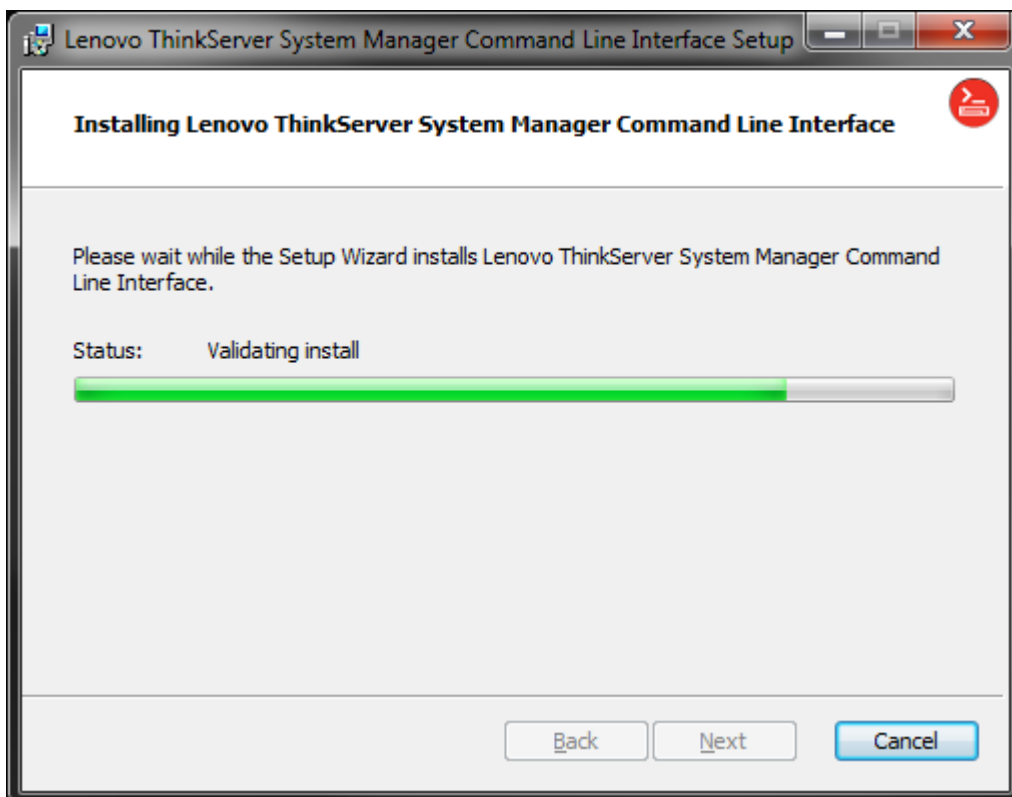
## 4. Select the location where the TSMCLI will be installed and also which shortcuts will be created.



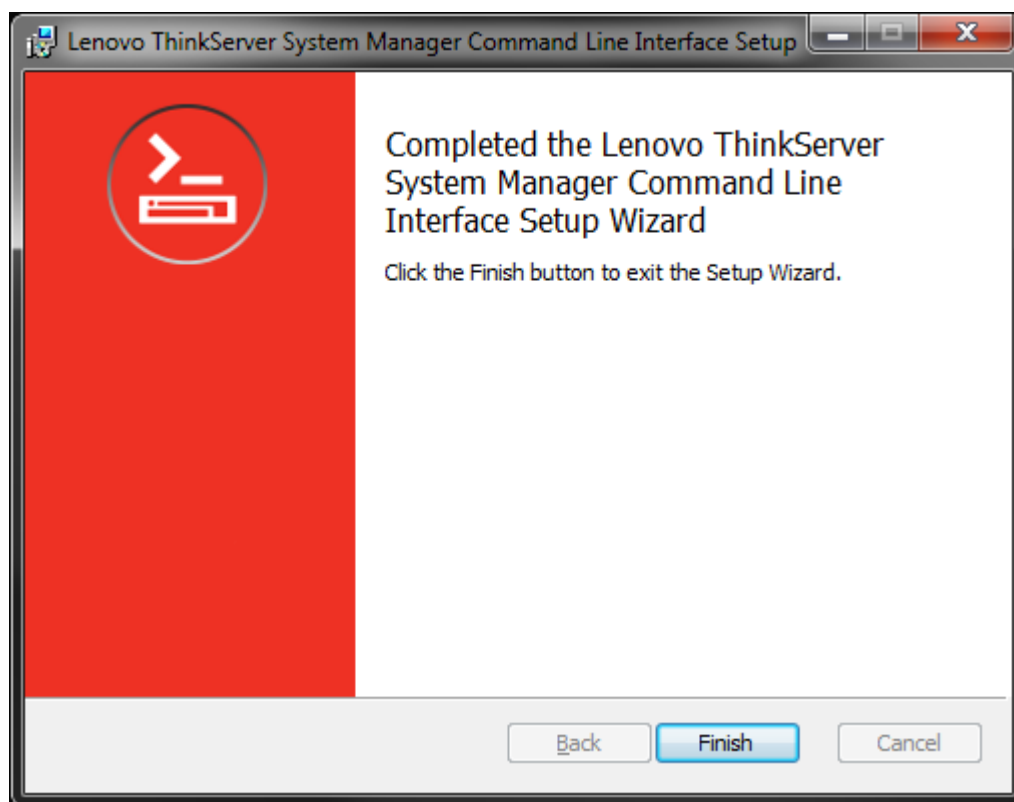
5. Click **Install** to confirm your preferences and install the TSMCLI.



6. Wait until the installation finishes.



7. Click on **Finish** to close the installer window.



**\*Install the TSMCLI using Windows command prompt:**

1. Start command prompt with **Administrator privilege**.
2. Change to directory of TSMCLI installer.
3. Confirm the file name of the installer.(For example: TSMCLIWindowsInstaller.msi)
4. Run one of the two following commands(depends on the requirement):

Install to default directory:

- TSMCLIWindowsInstaller.msi /q

Install to specified directory(For example: C:\Lenovo\TSMCLI):

- TSMCLIWindowsInstaller.msi INSTALLFOLDER="C:\Lenovo\TSMCLI" /q

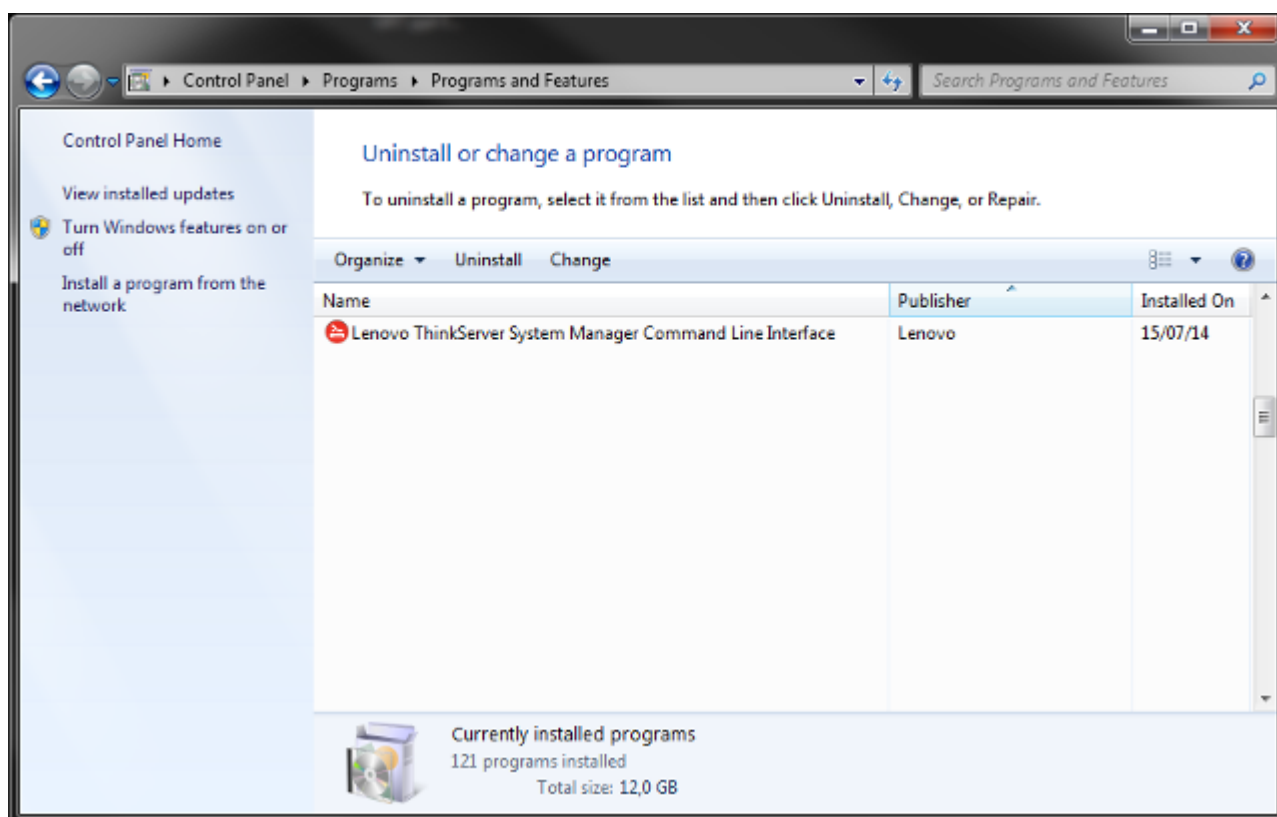
5. Wait until the installation finishes.

## 4. Uninstall Procedure

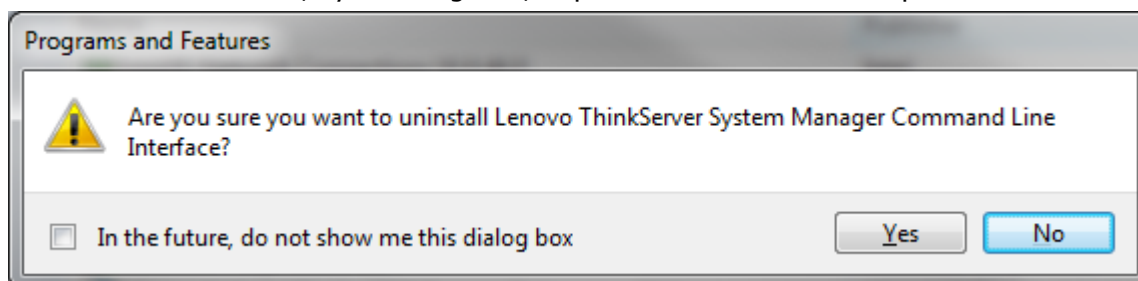
In order to uninstall the TSMCLI, make sure you have Windows **Administrator Privileges** and follow the instructions below:

### Uninstall the TSMCLI using Windows Control Panel:

1. Close all running instances of the TSMCLI.
2. Go to **Add/Remove Programs (Programs and Features** in Windows 7) located in the **Control Panel**.
3. Search for **Lenovo ThinkServer System Manager Command Line Interface**.
4. Click on the **Uninstall** button.



5. Confirm the uninstall, by selecting **Yes**, to proceed with the uninstall process.



6. Wait until the uninstall process finishes.

**Uninstall the TSMCLI using Windows command prompt:**

1. Close all running instances of the TSMCLI.
2. Start command prompt with **Administrator privilege**.
3. Run one of the two following commands(depends on the version installed):

For x86 version:

- wmic product where name="Lenovo ThinkServer System Manager Command Line Interface (x86)" call uninstall

For x64 version:

- wmic product where name="Lenovo ThinkServer System Manager Command Line Interface" call uninstall

4. Wait until the uninstall process finishes.

## 5. Usage

The basic TSMCLI command line syntax is the following:

```
Invoke-TSMCLI [-CommandName] <COMMAND_NAME> [COMMAND_PARAMETERS] [[-CommandArgs] @{COMMAND_ARGUMENTS}]
```

Where:

- `<COMMAND_NAME>` is the TSMCLI command which will be executed. Examples: "get-help", "get-users", "set-nicsettings".  
For the complete list of commands available check the [Chapter 6, TSMCLI Commands Reference](#) on this document.
- `[COMMAND_PARAMETERS]` is one of the following parameters:
  - ComputerName  
Specifies the Lenovo ThinkServer System Manager IP address or network name.
  - Port  
Specifies the Lenovo ThinkServer System Manager port number. This value is optional and if not specified will be set to "80" (or "443" if -UseSSL is specified).
  - Authentication  
Specifies the authentication type to be used upon the connection to a Lenovo ThinkServer System Manager. This value is optional and will be set to "basic" if not specified.
  - Credential  
Specifies a file or structure containing credentials. It should be created using the "create-credential" command.
  - UseSSL  
Specifies that the Secure Sockets Layer (SSL) protocol should be used to establish a connection to the specified TSM. By default, SSL is not used.
  - SkipSSLCertificateCheck  
Specifies that when connecting using SSL, there will be no verification if the TSM certificate is valid. Use this option only when the remote TSM is trusted by other means, for example, if the TSM is part of a network that is physically secure and isolated.
  - LogLevel  
Specifies the maximum message level that should be logged. The supported values are: 0 - none (no log); 1 - fatal; 2 - critical; 3 - error; 4 - warning; 5 - notice; 6 - information; and 7 - debug. If not specified the default level is 6 - information.

### Note:

From version 4.10, TSM only permits SSL connection, so UseSSL parameter must be specified when connecting TSM of 4.10 or later.

- `[@{COMMAND_ARGUMENTS}]` specifies the arguments specific to each TSMCLI command. For more information check the [Chapter 6, TSMCLI Commands Reference](#) on this document.

Examples for each TSMCLI command can be found on [Chapter 6, TSMCLI Commands Reference](#).

## 6. Commands Output View

Some TSMCLI commands returns data as an array, and where you are expecting potentially lots more text the PowerShell will replaces it with a single ellipsis. However, we can change the way the output is displayed by using the Format cmdlet.

```

Lenovo ThinkServer System Manager Command Line Interface
Copyright (C) 2014 Lenovo. All rights reserved.
PS C:\Users\alourenco> Invoke-TSMCLI get-fwversions -ComputerName 10.35.101.142 -Credential admin.xml
FWImages
@{DeviceType=; DeviceStatus=; StatusFlag=; Reboot=; CurrentImageVersion=1.42.78800; NewImageVersion=1.42.78800}, @...
PS C:\Users\alourenco> _
    
```

Each format cmdlet has default properties that will be used if you do not specify specific properties to display. Each cmdlet also uses the same parameter name, Property, to specify which properties you want to display. In addition, we recommend save the the command output into a variable (e.g. \$obj) and use it as an input to the format cmdlet. See the following sample:

```
$obj=Invoke-TSMCLI get-fwversions -ComputerName myServer -Credential
credential.xml
```

### 1. Format-List:

```

Lenovo ThinkServer System Manager Command Line Interface
Copyright (C) 2014 Lenovo. All rights reserved.
PS C:\Users\alourenco> $obj=Invoke-TSMCLI get-fwversions -ComputerName 10.35.101.142 -Credential admin.xml
PS C:\Users\alourenco> $obj.FWImages | Format-List

DeviceType       : @(Code=1; Type=System Manager)
DeviceStatus     : @(Code=1; Status=Device present)
StatusFlag       : @(Code=3; StatusFlag=Not required)
Reboot           : @(Code=5; Reboot=The TSM will need to be rebooted during the update process)
CurrentImageVersion : 1.42.78800
NewImageVersion  : 1.42.78800

DeviceType       : @(Code=2; Type=BIOS)
DeviceStatus     : @(Code=1; Status=Device present)
StatusFlag       : @(Code=0; StatusFlag=Matches in bundle)
Reboot           : @(Code=2; Reboot=The update will take effect only after the Host is rebooted)
CurrentImageVersion : 1.43.0
NewImageVersion  : Not available

DeviceType       : @(Code=4; Type=PSU)
DeviceStatus     : @(Code=1; Status=Device present)
StatusFlag       : @(Code=0; StatusFlag=Matches in bundle)
Reboot           : @(Code=0; Reboot=No reboot required)
CurrentImageVersion : 1.93.0
NewImageVersion  : Not available

DeviceType       : @(Code=201; Type=Deployment Manager)
DeviceStatus     : @(Code=3; Status=Device present)
StatusFlag       : @(Code=0; StatusFlag=Matches in bundle)
Reboot           : @(Code=0; Reboot=No reboot required)
CurrentImageVersion : 1.2.5
NewImageVersion  : Not available
    
```

```
$obj.FWImages | Format-List
```

### 2. Format-Table:

```

Lenovo ThinkServer System Manager Command Line Interface
Copyright (C) 2014 Lenovo. All rights reserved.
PS C:\Users\alourenco> $obj=Invoke-TSMCLI get-fwversions -ComputerName 10.35.101.142 -Credential admin.xml
PS C:\Users\alourenco> $obj.FWImages | Format-Table -Property DeviceType, DeviceStatus, CurrentImageVersion

DeviceType       DeviceStatus       CurrentImageVersion
-----
@(Code=1; Type=System Manager)  @(Code=1; Status=Device present)  1.42.78800
@(Code=2; Type=BIOS)            @(Code=1; Status=Device present)  1.43.0
@(Code=4; Type=PSU)            @(Code=1; Status=Device present)  1.93.0
@(Code=201; Type=Deployment Manager)  @(Code=3; Status=Device present)  1.2.5
@(Code=202; Type=Windows Driver Bundle)  @(Code=3; Status=Device present)  1.2.4
@(Code=203; Type=Linux Driver Bundle)    @(Code=3; Status=Device present)  1.2.4
@(Code=205; Type=Diagnostic)          @(Code=3; Status=Device present)  1.0.59
    
```



```
$obj.FWImages | Format-Table -Property DeviceType, DeviceStatus,  
CurrentImageVersion
```

For further detail about Format cmdlet, please check <https://technet.microsoft.com/en-us/library/dd347677.aspx>. In addition, in order to get more details about each command return type and their properties, such as `$obj.FWImages`, please refer to [Commands Reference](#) section, and its RESPONSE subsection, in this document.

## 7. Commands Reference

### backup-config

Backs up the configuration of a specific TSM to a local file.

#### SYNTAX

```
Invoke-TSMCLI [-CommandName] backup-config -ComputerName <ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>] -Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-CommandArgs] @ {configFile="<config_file_path>"} [-LogLevel <0|1|2|3|4|5|6|7>]
```

#### DESCRIPTION

The backup-config command backs up the configuration of a specific Lenovo ThinkServer System Manager to a local file.

#### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

#### COMMAND ARGUMENTS

##### configFile

Specifies the filename and path where the backup will be saved (required).

#### EXAMPLES

```
Invoke-TSMCLI -CommandName backup-config -ComputerName myserver -Port 80 -Authentication basic -Credential $cred -CommandArgs @ {configFile="C:\myconfigfile"}
```

#### RESPONSE

Object Result		
Integer	<b>Code</b>	The response code for "Success" (always 0).
String	<b>Message</b>	The localized message for "Success".

## clear-sel

Clears the SEL (System Event Log) of a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] clear-sel -ComputerName <ComputerName> [-Port
<PortNumber>] [-Authentication <AuthenticationMode>] -Credential
<credential> [-UseSSL [-SkipSSLCertificateCheck]] [-LogLevel <0|1|2|3|4|5|
6|7>]
```

### DESCRIPTION

The clear-sel command clears the SEL of the specified Lenovo ThinkServer System Manager.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### EXAMPLES

```
Invoke-TSMCLI -CommandName clear-sel -ComputerName myserver -Port 80 -
Authentication basic -Credential $credential
```

### RESPONSE

Object Result		
Integer	<b>Result</b>	The response code for "Success" (always 0).
String	<b>Message</b>	The localized message for "Success".

## create-adrolegroup

Creates an Active Directory role group on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] create-adrolegroup -ComputerName  
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]  
-Credential <credential> [-UseSSL [-SkipSSLCertificateCheck]] [-  
CommandArgs] @  
{groupName="<group_name>";domainName="<domain_name>";role="<admin|user|  
operator>" [;kvm="<0|1>"] [;virtualMedia="<0|1>"]} [-LogLevel <0|1|2|3|4|5|  
6|7>]
```

### DESCRIPTION

The create-adrolegroup command creates an Active Directory role group on a specific Lenovo ThinkServer System Manager. This role group maps an Active Directory group to a TSM role, and is used to control which access level is granted to this group.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### groupName

Name of the Active Directory group, with a maximum of 63 characters. Hyphens and underscores are allowed.

#### domainName

Name of the Active Directory domain, with a maximum of 255 characters. A domain name must contain at least two labels (separated by a ".") and must not start with a ".". Each label can be up to 63 characters long. Hyphens are allowed, but not at the start or end of a label. Underscores are allowed, but not at the start of a label. Examples: "my2company.com" and "my-network.localdomain".

#### role

Specifies the group role. Possible values are: "admin" for Administrator, "user" for User or "operator" for Operator.

#### kvm

Specifies the KVM privilege (optional). Enabled by default for "admin" and "operator" roles.

#### virtualMedia

Specifies the Virtual Media privilege (optional). Enabled by default for "admin" and "operator" roles.

## EXAMPLES

```
Invoke-TSMCLI -CommandName create-adrolegroup -ComputerName myserver -Port
80 -Authentication basic -Credential $cred -CommandArgs @
{groupName="DomainOperators";domainName="mycompany.com";role="operator";kv
m="0";virtualMedia="1" }
```

## RESPONSE

Object Result		
Integer	<b>Result</b>	The response code for "Success" (always 0).
String	<b>Message</b>	The localized message for "Success".
Integer	<b>RoleGroupId</b>	The ID of the newly created AD role group.

## create-alertnotification

Creates a new Alert Notification on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] create-alertnotification -ComputerName
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]
-Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-
CommandArgs] @{ipAddress="<ip address>";powerAction="<none|power-down|
power-reset|power-cycle>";<SENSOR ARGUMENTS>} [-LogLevel <0|1|2|3|4|5|6|
7>]
```

SENSOR ARGUMENTS

```
sensorMode="all"
```

```
sensorMode="sensor-type";sensorType="<sensor_type>"
```

### DESCRIPTION

The create Alert Notification command is a bulk command that execute all the necessary steps to enable a new alert notification for the TSM. This bulk command create a new SNMP trap, a new Alert Policy and a new Event Filter, in order to create a new alert notification.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### ipAddress

The IPv4 or IPv6 address to where SNMP notifications will be sent (required). Provided IPv4 addresses must not start with 0.

#### powerAction

Selects the power action to be executed (required). It can be "none", "power-down", "power-reset" or "power-cycle".

#### sensorMode

The sensor selection mode (required). It can be:

all - to select all sensors;

sensor-type - to select sensors by type ID;

#### sensorType

The sensor type identification (required if sensorMode is "sensor-type"). To get the list of available sensor types, invoke the TSMCLI command "get-sensortypes".

## EXAMPLES

```
Invoke-TSMCLI -CommandName create-alertnotification -ComputerName myserver
-Port 80 -Authentication basic -Credential $cred -CommandArgs @
{ipAddress="192.168.1.10";powerAction="none";sensorMode="all"}
```

```
Invoke-TSMCLI -CommandName create-alertnotification -ComputerName myserver
-Port 80 -Authentication basic -Credential $cred -CommandArgs @
{ipAddress="192.168.1.10";powerAction="power-down";sensorMode="sensor-
type";sensorType=4}
```

```
Invoke-TSMCLI -CommandName create-alertnotification -ComputerName myserver
-Port 80 -Authentication basic -Credential $cred -CommandArgs @
{ipAddress="192.168.1.10";powerAction="power-reset";sensorMode="sensor-
type";sensorType=4}
```

```
Invoke-TSMCLI -CommandName create-alertnotification -ComputerName myserver
-Port 80 -Authentication basic -Credential $cred -CommandArgs @
{ipAddress="192.168.1.10";powerAction="power-cycle";sensorMode="all"}
```

## RESPONSE

Object Result		
Integer	<b>Result</b>	The response code for "Success" (always 0).
String	<b>Message</b>	The localized message for "Success".

## create-alertpolicy

Creates a new Alert Policy on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] create-alertpolicy -ComputerName  
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]  
-Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-  
CommandArgs] @{policyNumber=<policy number>;policyEnabled=<0|  
1>;policySet=<policy set  
number>;lanDestinationId=<destination>;eventSpecificString=<0|  
1>;alertStringKey=<alert string number>} [-LogLevel <0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The create-alertpolicy command creates a new alert policy on a specific Lenovo ThinkServer System Manager.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### policyNumber

The policy number that will be used in event filters (required). Valid values are 1 to 15.

#### policyEnabled

Indicates if the Policy Alert should be enabled (1) or not (0) (required).

#### policySet

The policy set number (required). It can be:

0 - Always send alert to this destination;

1 - If alert to previous destination was successful, do not send alert to this destination. Proceed to next entry in this policy set;

2 - If alert to previous destination was successful, do not send alert to this destination. Do not process any more entries in this policy set;

3 - If alert to previous destination was successful, do not send alert to this destination. Proceed to next entry in this policy set that is to a different channel;

4 - If alert to previous destination was successful, do not send alert to this destination. Proceed to next entry in this policy set that is to a different destination type.

#### lanDestinationId



The LAN destination of the policy (required). To get the list of available LAN destinations, invoke the TSMCLI command "get-landestinations".

**eventSpecificString**

Indicates if an event specific alert string should be used (1) or not (0) (required).

**alertStringKey**

The alert string key contained in the PEF configuration parameters, to specify which string is to be sent for this Alert Policy (required). Valid values are between the range 0 to 127.

**EXAMPLES**

```
Invoke-TSMCLI -CommandName create-alertpolicy -ComputerName myserver -Port
80 -Authentication basic -Credential $cred -CommandArgs @
{policyNumber=3;policyEnabled=1;policySet=2;lanDestinationId=4;eventSpecif
icString=1;alertStringKey=15}
```

**RESPONSE**

Object Result		
Integer	<b>Result</b>	The response code for "Success" (always 0).
String	<b>Message</b>	The localized message for "Success".
Integer	<b>AlertPolicyId</b>	The ID of the newly created Alert Policy.

## create-credential

Creates a structure containing user credentials.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] create-credential [[-CommandArgs] @ {[username = "<name>";] [password = "<password>";] [saveToFile = "<file>"]}] [-LogLevel <0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The create-credential command stores user credentials in a structure that can be passed to subsequent TSMCLI calls.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### username

Specifies the username. In the absence of this argument, an error message is shown.

#### password

Specifies the password. In the absence of this argument, an error message is shown.

#### saveToFile

Specifies the name of the file where to save the structure containing the username and password.

### EXAMPLES

```
$cred = Invoke-TSMCLI -CommandName create-credential -CommandArgs @ {username="user"; password = "pass"}
```

```
Invoke-TSMCLI -CommandName create-credential -CommandArgs @ {username="user"; password = "pass"; saveToFile="credentials.xml"}
```

### RESPONSE

Object Result	
String	<b>Username</b> The username used in the credential.
String	<b>Password</b> The encrypted password used in the credential.

## create-eventfilter

Creates a new Event Filter on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] create-eventfilter -ComputerName
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]
-Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-
CommandArgs] @{eventFilterEnabled="<0|1>";severity="<unspecified|monitor|
information|normal|non-critical|critical|non-
recoverable>";powerAction="<none|power-down|power-reset|power-
cycle>";policyNumber="<1-15>";<GENERATOR ID ARGUMENTS>;<SENSOR
ARGUMENTS>;<EVENT DATA ARGUMENTS>} [-LogLevel <0|1|2|3|4|5|6|7>]
```

#### GENERATOR ID ARGUMENTS

There are 3 different ways of specifying the generatorId arguments:

1. By setting generatorIdRawData="1" and also specifying generatorId1="<hex\_value>;generatorId2="<hex\_value>"
2. By setting generatorIdRawData="0" and eventGenerator="slave-address" and also specifying generator="<slave\_address>;generatorChannelNumber="<generator\_channel\_number>;generatorIPMBDeviceLUN="<generator\_ipmb\_device\_lun>"
3. By setting generatorIdRawData="0" and eventGenerator="software-id" and also specifying generator="<software\_id>;generatorChannelNumber="<generator\_channel\_number>"

All three above methods produce values for generatorId1 and generatorId2, which will be displayed by the TSMCLI command "get-eventfilters" when visualizing the information of the corresponding event filter.

#### SENSOR ARGUMENTS

```
sensorMode="all"
```

```
sensorMode="sensor-type";sensorType="<sensor_type>;eventOptions="all"
```

```
sensorMode="sensor-type";sensorType="<sensor_type>;eventOptions="sensor-
events";sensorEvents="<sensor-event-1,sensor-event-2,...,sensor-event-n>"
```

```
sensorMode="sensor-name";sensorName="<sensor_name>;eventOptions="all"
```

```
sensorMode="sensor-name";sensorName="<sensor_name>;eventOptions="sensor-
events";sensorEvents="<sensor-event-1,sensor-event-2,...,sensor-event-n>"
```

#### EVENT DATA ARGUMENTS

```
eventTrigger="<event_trigger>"[;eventData1ANDMask="<event_data_1_and_mask>
"][;eventData1Compare1="<event_data_1_compare_1>"]
[;eventData1Compare2="<event_data_1_compare_2>"]
[;eventData2ANDMask="<event_data_2_and_mask>"]
[;eventData2Compare1="<event_data_2_compare_1>"]
[;eventData2Compare2="<event_data_2_compare_2>"]
[;eventData3ANDMask="<event_data_3_and_mask>"]
[;eventData3Compare1="<event_data_3_compare_1>"]
[;eventData3Compare2="<event_data_3_compare_2>"]
```

## DESCRIPTION

The `create-eventfilter` command creates a new platform event filter (PEF) entry to select which event should be alerted to users on a specific Lenovo ThinkServer System Manager.

## PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

## COMMAND ARGUMENTS

### **eventFilterEnabled**

Indicates if the Event Filter should be enabled (1) or not (0) (required).

### **severity**

The minimum event severity to filter (required). The Event Filter severity can be "unspecified", "monitor", "information", "normal", "non-critical", "critical" or "non-recoverable".

### **powerAction**

Selects the power action to be executed (required). It can be "none", "power-down", "power-reset" or "power-cycle".

### **policyNumber**

Specifies the policy number related to this event filter (required). It allows to trigger alert policies associated with this policy number. Valid values are 1 to 15.

### **generatorIDRawData**

Indicates if the Event Filter should use a generator from raw data (1) or from specified generator (0) (required).

### **eventGenerator**

Indicates if the event generator should be an I2C slave address or a software ID (required if `generatorIDRawData` is "0"). It can be "slave-address" or "software-id".

### **generator**

The generator I2C address or software ID (required if `generatorIDRawData` is "0"). Valid values are 0 to 127.

### **generatorChannelNumber**

The particular channel number through which the event message is received over (required if generatorIDRawData is "0"). Choose '0' if the event message is received via the system interface, primary IPMB, or internally generated by the TSM. Valid values are 0 to 15.

**generatorIPMBDeviceLUN**

The corresponding IPMB device LUN if event generated by IPMB (required if eventGenerator is "slave-address"). Valid values are 1 to 3.

**generatorID1**

The raw generator ID1 data value (required if generatorIDRawData is "1"). Valid values are 0x0 to 0xFF.

**generatorID2**

The raw generator ID2 data value (required if generatorIDRawData is "1"). Valid values are 0x0 to 0xFF.

**sensorMode**

The sensor selection mode (required). It can be:

all - to select all sensors;

sensor-type - to select sensors by type ID;

sensor-name - to select sensors by name.

**sensorType**

The sensor type identification (required if sensorMode is "sensor-type"). To get the list of available sensor types, invoke the TSMCLI command "get-sensortypes".

**sensorName**

The sensor name (required if sensorMode is "sensor-name"). To get the list of available sensor names, invoke the TSMCLI command "get-sensors".

**eventOptions**

Specifies the events from the selected sensors (required if sensorMode is different from "all"). Possible values are: "all" for all events, "sensor-events" for sensor specific events.

**sensorEvents**

A comma separated list of all possible events for the selected sensors (required if eventOptions is "sensor-events"). Each event can be:

lower-non-critical-going-low;

lower-non-critical-going-high;

lower-critical-going-low;

lower-critical-going-high;

lower-non-recoverable-going-low;

lower-non-recoverable-going-high;

upper-non-critical-going-low;

upper-non-critical-going-high;

upper-critical-going-low;

upper-critical-going-high;

upper-non-recoverable-going-low;

upper-non-recoverable-going-high.

**eventTrigger**

This event/reading type value (required). Valid values are 1 to 255.

**eventData1ANDMask**

Indicates wildcarded or compared bits. Valid values are 0 to 255.

**eventData1Compare1**

Indicates whether each bit position's comparison is an exact comparison or not. Valid values are 0 to 255.

**eventData1Compare2**

Indicates whether each bit position's comparison is an exact comparison or not. Valid values are 0 to 255.

**eventData2ANDMask**

Indicates wildcarded or compared bits. Valid values are 0 to 255.

**eventData2Compare1**

Indicates whether each bit position's comparison is an exact comparison or not. Valid values are 0 to 255.

**eventData2Compare2**

Indicates whether each bit position's comparison is an exact comparison or not. Valid values are 0 to 255.

**eventData3ANDMask**

Indicates wildcarded or compared bits. Valid values are 0 to 255.

**eventData3Compare1**

Indicates whether each bit position's comparison is an exact comparison or not. Valid values are 0 to 255.

**eventData3Compare2**

Indicates whether each bit position's comparison is an exact comparison or not. Valid values are 0 to 255.

**EXAMPLES**

```
Invoke-TSMCLI -CommandName create-eventfilter -ComputerName myserver -Port
80 -Authentication basic -Credential $cred -CommandArgs @
{eventFilterEnabled=1;severity="non-
critical";powerAction="none";policyNumber=4;generatorIdRawData="1";generat
```

```
orId1="0xFF";generatorId2="0xFF";sensorMode="sensor-
name";sensorName="Fan_4-1";eventOptions="all";eventTrigger=255;eventData1A
NDMask=0;eventData1Compare1=0;eventData1Compare2=0;eventData2ANDMask=2;eve
ntData2Compare1=2;eventData2Compare2=0;eventData3ANDMask=1;eventData3Compa
rel=1;eventData3Compare2=1}
```

```
Invoke-TSMCLI -CommandName create-eventfilter -ComputerName myserver -Port
80 -Authentication basic -Credential $cred -CommandArgs @
{eventFilterEnabled=1;severity="non-
critical";powerAction="none";policyNumber=4;generatorIdRawData="1";generat
orId1="0xFF";generatorId2="0xFF";sensorMode="sensor-
name";sensorName="Fan_4-1";eventOptions="sensor-
events";sensorEvents="lower-non-critical-going-low,lower-non-critical-
going-high,upper-non-critical-going-
low";eventTrigger=255;eventData1ANDMask=0;eventData1Compare1=0;eventData1C
ompare2=0;eventData2ANDMask=2;eventData2Compare1=2;eventData2Compare2=0;ev
entData3ANDMask=1;eventData3Compare1=1;eventData3Compare2=1}
```

```
Invoke-TSMCLI -CommandName create-eventfilter -ComputerName myserver -Port
80 -Authentication basic -Credential $cred -CommandArgs @
{eventFilterEnabled=1;severity="non-
critical";powerAction="none";policyNumber=4;generatorIdRawData="1";generat
orId1="0xFF";generatorId2="0xFF";sensorMode="sensor-
type";sensorType=4;eventOptions="all";eventTrigger=255;eventData1ANDMask=
0;eventData1Compare1=0;eventData1Compare2=0;eventData2ANDMask=2;eventData2
Compare1=2;eventData2Compare2=0;eventData3ANDMask=1;eventData3Compare1=1;e
ventData3Compare2=1}
```

```
Invoke-TSMCLI -CommandName create-eventfilter -ComputerName myserver -Port
80 -Authentication basic -Credential $cred -CommandArgs @
{eventFilterEnabled=1;severity="non-critical";powerAction="power-
down";policyNumber=4;generatorIdRawData="0";eventGenerator="slave-
address";generator="2";generatorChannelNumber=0;generatorIPMBDeviceLUN=1;s
ensorMode="all";eventTrigger=255;eventData1ANDMask=0;eventData1Compare1=0;
eventData1Compare2=0;eventData2ANDMask=2;eventData2Compare1=2;eventData2Co
mpare2=0;eventData3ANDMask=1;eventData3Compare1=1;eventData3Compare2=1}
```

```
Invoke-TSMCLI -CommandName create-eventfilter -ComputerName myserver -Port
80 -Authentication basic -Credential $cred -CommandArgs @
{eventFilterEnabled=1;severity="non-critical";powerAction="power-
down";policyNumber=4;generatorIdRawData="0";eventGenerator="slave-
address";generator="2";generatorChannelNumber=0;generatorIPMBDeviceLUN=1;s
ensorMode="all";eventTrigger=255;eventData1ANDMask=0;eventData1Compare1=0;
eventData1Compare2=0;eventData2ANDMask=2;eventData2Compare1=2;eventData2Co
mpare2=0;eventData3ANDMask=1;eventData3Compare1=1;eventData3Compare2=1}
```

```
Invoke-TSMCLI -CommandName create-eventfilter -ComputerName myserver -Port
80 -Authentication basic -Credential $cred -CommandArgs @
{eventFilterEnabled=1;severity="non-
critical";powerAction="none";policyNumber=4;generatorIdRawData="1";generat
orId1="0xFF";generatorId2="0xFF";eventGenerator="software-
id";generator="1";generatorChannelNumber=0;sensorMode="all";eventTrigger=2
55;eventData1ANDMask=0;eventData1Compare1=0;eventData1Compare2=0;eventData
2ANDMask=2;eventData2Compare1=2;eventData2Compare2=0;eventData3ANDMask=1;e
ventData3Compare1=1;eventData3Compare2=1}
```

## RESPONSE

Object Result		
Integer	<b>Result</b>	The response code for <b>Success</b> (always 0).
String	<b>Message</b>	The localized message for <b>Success</b> .
Integer	<b>EventFilterId</b>	The ID of the newly created Event Filter.



## create-firewalliprule

Adds a firewall rule for blocking access to a specific TSM from an IP address or an IP address range.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] create-firewalliprule -ComputerName
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]
-Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-
CommandArgs] @{ip="<ipv4_address>"} [-LogLevel <0|1|2|3|4|5|6|7>]
```

```
Invoke-TSMCLI [-CommandName] create-firewalliprule -ComputerName
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]
-Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-
CommandArgs] @{ipStart="<ipv4_address>"; ipEnd="<ipv4_address>"} [-
LogLevel <0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The create-firewalliprule command adds a rule for blocking access from a single IP address or a range of IP addresses to a specific Lenovo ThinkServer System Manager. Note that the TSM allows the creation of identical and overlapping rules and each individual rule is managed separately. If the same IP address is blocked by multiple rules, all of them will have to be deleted for connections from that IP address to be allowed.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### ip

Specifies a single IP address. It's not used if a range of IP addresses is being specified. Only IPv4 address format is supported.

#### ipStart

Specifies the start IP of an IP address range. It's not used if a single IP address is being specified. Only IPv4 address format is supported.

#### ipEnd

Specifies the end IP of an IP address range. It's not used if a single IP address is being specified. Only IPv4 address format is supported.

### EXAMPLES

```
Invoke-TSMCLI -CommandName create-firewalliprule -ComputerName myserver -
Port 80 -Authentication basic -Credential $cred -CommandArgs @
{ip="192.168.1.104"}
```

```
Invoke-TSMCLI -CommandName create-firewalliprule -ComputerName myserver -  
Port 80 -Authentication basic -Credential $cred -CommandArgs @  
{ipStart="192.168.1.104"; ipEnd="192.168.1.108"}
```

## RESPONSE

Object		Result
Integer	<b>Result</b>	The response code for "Success" (always 0).
String	<b>Message</b>	The localized message for "Success".

## create-firewallportrule

Adds a rule for blocking access to a single port or a port range of a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] create-firewallportrule -ComputerName
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]
-Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-
CommandArgs] @{protocol="<tcp|udp>"; port="<port number>"} [-LogLevel <0|
1|2|3|4|5|6|7>]
```

```
Invoke-TSMCLI [-CommandName] create-firewallportrule -ComputerName
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]
-Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-
CommandArgs] @{protocol="<tcp|udp>"; portStart="<port number>";
portEnd="<port number>"} [-LogLevel <0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The create-firewallportrule command adds a rule for blocking access to a single port or a port range of a specific Lenovo ThinkServer System Manager. Note that the TSM allows the creation of identical and overlapping rules and each individual rule is managed separately. If the same port is blocked by multiple rules, all of them will have to be deleted for connections from that port to be allowed.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### protocol

The protocol type. It can be "tcp" or "udp".

#### port

Specifies a single port number. It's not used if a port range is being specified. Valid values are 1 to 65535.

#### portStart

Specifies a port range start. It's not used if a single port is being specified. Valid values are 1 to 65534.

#### portEnd

Specifies a port range end. It's not used if a single port is being specified. Valid values are 2 to 65535.

## EXAMPLES

```
Invoke-TSMCLI -CommandName create-firewallportrule -ComputerName myserver
-Port 80 -Authentication basic -Credential $cred -CommandArgs @
{protocol="tcp"; port="50000"}
```

```
Invoke-TSMCLI -CommandName create-firewallportrule -ComputerName myserver
-Port 80 -Authentication basic -Credential $cred -CommandArgs @
{protocol="udp"; portStart="50000"; portEnd="50100"}
```

## RESPONSE

Object Result		
Integer	<b>Result</b>	The response code for "Success" (always 0).
String	<b>Message</b>	The localized message for "Success".

## create-landestination

Creates a new LAN Destination on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] create-landestination -ComputerName
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]
-Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-
CommandArgs] @{type="snmp";destinationAddress="<ip address>"} [-LogLevel
<0|1|2|3|4|5|6|7>]
```

```
Invoke-TSMCLI [-CommandName] create-landestination -ComputerName
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]
-Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-
CommandArgs] @{type="email";userId="<user_id>"} [-LogLevel <0|1|2|3|4|5|6|
7>]
```

### DESCRIPTION

The create-landestination command creates a new LAN Destination (SNMP trap or email) that will be used by platform event filters (PEF) to send notifications from a specific Lenovo ThinkServer System Manager.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### type

Specifies the type of LAN destination (required). The possible values are "snmp" if the destination is snmp trap or "email" if it is an e-mail.

#### destinationAddress

The IPv4 or IPv6 address to where SNMP notifications will be sent (required if type is "snmp"). Provided IPv4 addresses must not start with 0.

#### userId

The user to whom email alerts will be sent (required if type is "email"). Note that an email address must be configured for the selected user account. To get the registered users, invoke the TSMCLI command "get-users".

### EXAMPLES

```
Invoke-TSMCLI -CommandName create-landestination -ComputerName myserver -
Port 80 -Authentication basic -Credential $cred -CommandArgs @
{type="snmp";destinationAddress="192.168.1.10"}
```

```
Invoke-TSMCLI -CommandName create-landestination -ComputerName myserver -  
Port 80 -Authentication basic -Credential $cred -CommandArgs @  
{type="email";userId=2}
```

## RESPONSE

Object		Result
Integer	<b>Result</b>	The response code for "Success" (always 0).
String	<b>Message</b>	The localized message for "Success".
Integer	<b>LANDestinationId</b>	The ID of the newly created LAN Destination.

## create-ldaprolegroup

Creates an LDAP role group on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] create-ldaprolegroup -ComputerName  
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]  
-Credential <credential> [-UseSSL [-SkipSSLCertificateCheck]] [-  
CommandArgs] @  
{groupName="<group_name>";groupSearchBase="<group_search_base>";role="<adm  
in|user|operator>"[;kvm="<0|1>"}[;virtualMedia="<0|1>"} [-LogLevel <0|1|  
2|3|4|5|6|7>]
```

### DESCRIPTION

The create-ldaprolegroup command creates an LDAP role group on a specific Lenovo ThinkServer System Manager. This role group maps an LDAP group to a TSM role, and is used to control which access level is granted to this group.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### groupName

Name of the LDAP group, with a maximum of 63 characters. Hyphens and underscores are allowed.

#### groupSearchBase

Search base for groups on the LDAP server. It shall be at least 4 and at most 63 alphanumeric characters long. Hyphens, dots and underscores are allowed. Examples are "dc=corp,dc=domain,dc=com" and "dc=people,dc=mydept,dc=mycompany,dc=com".

#### role

Specifies the group role. Possible values are: "admin" for Administrator, "user" for User or "operator" for Operator.

#### kvm

Specifies the KVM privilege (optional). Enabled by default for "admin" and "operator" roles.

#### virtualMedia

Specifies the Virtual Media privilege (optional). Enabled by default for "admin" and "operator" roles.

## EXAMPLES

```
Invoke-TSMCLI -CommandName create-ldaprolegroup -ComputerName myserver -
Port 80 -Authentication basic -Credential $cred -CommandArgs @
{groupName="DomainOperators";groupSearchBase="dc=corp,dc=domain,dc=com";ro
le="operator";kvm="1";virtualMedia="0"}
```

## RESPONSE

Object Result		
Integer	<b>Result</b>	The response code for "Success" (always 0).
String	<b>Message</b>	The localized message for "Success".
Integer	<b>RoleGroupId</b>	The ID of the newly created LDAP role group.



## create-user

Creates a new user account on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] create-user -ComputerName <ComputerName> [-
Port <PortNumber>] [-Authentication <AuthenticationMode>] -Credential
<credential> [-UseSSL [-SkipSSLCertificateCheck]] [-CommandArgs] @
{username="<username>";password="<password>";role="<admin|user|
operator>" [;email="<email>"] [;kvm="<0|1>"] [;virtualMedia="<0|1>"]
[;snmp="<0|1>";snmpAccess="<read-only|read-
write>";authenticationProtocol="<sha|md5>";privacyProtocol="<des|aes>"} }
[-LogLevel <0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The create-user command creates a new user account on a specific Lenovo ThinkServer System Manager.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### username

Specifies the user name. It shall have a maximum of 16 characters, must start with an alphabetical character and is case sensitive. The following special characters are allowed: '-'(hyphen), '\_'(underscore) and '@'(at sign).

#### password

Specifies the password. It shall have a maximum of 20 characters and only printable ASCII characters and spaces are accepted. In addition, if SNMP is enabled, the password shall have at least 8 characters.

#### role

Specifies the user role. Possible values are: "admin" for Administrator, "user" for User or "operator" for Operator.

#### email

Specifies the user email (optional). The email length is limited to 63 characters and can be defined as empty string (""). In addition, the e-mail shall be in the format <username>@<domain>. Examples: "myname@mycompany.com" and "my.mail@dept.example.com".

#### kvm

Specifies the KVM privilege (optional). Enabled by default for "admin" and "operator" roles.

**virtualMedia**

Specifies the Virtual Media privilege (optional). Enabled by default for "admin" and "operator" roles.

**snmp**

Specifies the SNMP enabled state (optional).

**snmpAccess**

Specifies the SNMP access level (required if enabling SNMP). Possible values are: "read-only" or "read-write".

**authenticationProtocol**

Specifies the authentication protocol for SNMP settings (required if enabling SNMP). Possible values are: "sha" or "md5".

**privacyProtocol**

Specifies the encryption algorithm to use for SNMP settings (required if enabling SNMP). Possible values are: "des" or "aes".

**EXAMPLES**

```
Invoke-TSMCLI -CommandName create-user -ComputerName myserver -Port 80 -
Authentication basic -Credential $credential -CommandArgs @
{username="mynewuser";password="1234";role="user"}
```

```
Invoke-TSMCLI -CommandName create-user -ComputerName myserver -Port 80 -
Authentication basic -Credential $credential -CommandArgs @
{username="admin2";password="1234";role="admin";email="mynewadmin@mydomai
n.com";kvm="1";virtualMedia="1";snmp="1";snmpAccess="read-
only";authenticationProtocol="sha";privacyProtocol="des"}
```

**RESPONSE**

Object Result		
Integer	<b>Result</b>	The response code for "Success" (always 0).
String	<b>Message</b>	The localized message for "Success".
Integer	<b>UserId</b>	The ID of the newly created user.

## delete-adrolegroup

Deletes an Active Directory role group from a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] delete-adrolegroup -ComputerName
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]
-Credential <credential> [-UseSSL [-SkipSSLCertificateCheck]] [-
CommandArgs] @{roleGroupId=<role_group_id>} [-LogLevel <0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The delete-adrolegroup command deletes an Active Directory role group from a specific Lenovo ThinkServer System Manager.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### roleGroupId

Specifies the role group identifier. To get the list of valid user identifiers on the TSM, use the command "get-adrolegroups".

### EXAMPLES

```
Invoke-TSMCLI -CommandName delete-adrolegroup -ComputerName myserver -Port
80 -Authentication basic -Credential $credential -CommandArgs @
{roleGroupId=1}
```

### RESPONSE

		Object Result
Integer	<b>Result</b>	The response code for "Success" (always 0).
String	<b>Message</b>	The localized message for "Success".

## delete-alertpolicy

Deletes an Alert Policy on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] delete-alertpolicy -ComputerName
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]
-Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-
CommandArgs] @{alertPolicyId="alert_policy_id"} [-LogLevel <0|1|2|3|4|5|
6|7>]
```

### DESCRIPTION

The delete-alertpolicy command deletes an alert policy on a specific Lenovo ThinkServer System Manager.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### alertPolicyId

The id of the Alert Policy to be deleted. To get the list of valid alertPolicyIds on the TSM, use the command "get-alertpolicies".

### EXAMPLES

```
Invoke-TSMCLI -CommandName delete-alertpolicy -ComputerName myserver -Port
80 -Authentication basic -Credential $cred -CommandArgs @
{alertPolicyId="1"}
```

### RESPONSE

Object Result		
Integer	<b>Result</b>	The response code for "Success" (always 0).
String	<b>Message</b>	The localized message for "Success".

## delete-eventfilter

Deletes an Event Filter on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] delete-eventfilter -ComputerName
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]
-Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-
CommandArgs] @{eventFilterId="<<event-filter-id>"} [-LogLevel <0|1|2|3|4|5|
6|7>]
```

### DESCRIPTION

The delete-eventfilter command deletes an Event Filter on a specific Lenovo ThinkServer System Manager.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### eventFilterId

The id of the Event Filter to be deleted.

### EXAMPLES

```
Invoke-TSMCLI -CommandName delete-eventfilter -ComputerName myserver -Port
80 -Authentication basic -Credential $cred -CommandArgs @
{eventFilterId="5"}
```

### RESPONSE

Object Result		
Integer	<b>Result</b>	The response code for "Success" (always 0).
String	<b>Message</b>	The localized message for "Success".

## delete-firewalliprule

Deletes a firewall IP address or IP address range blocking rule on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] delete-firewalliprule -ComputerName  
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]  
-Credential <credential> [-UseSSL [-SkipSSLCertificateCheck]] [-  
CommandArgs] @{ip="<ipv4_address>"} [-LogLevel <0|1|2|3|4|5|6|7>]
```

```
Invoke-TSMCLI [-CommandName] delete-firewalliprule -ComputerName  
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]  
-Credential <credential> [-UseSSL [-SkipSSLCertificateCheck]] [-  
CommandArgs] @{ipStart="<ipv4_address>"; ipEnd="<ipv4_address>"} [-  
LogLevel <0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The delete-firewalliprule command deletes a firewall IP rule from a specific Lenovo ThinkServer System Manager.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### ip

Specifies a single IP address. It's not used if a range of IP addresses is being specified. Only IPv4 address format is supported.

#### ipStart

Specifies the start IP of an IP address range. It's not used if a single IP address is being specified. Only IPv4 address format is supported.

#### ipEnd

Specifies the end IP of an IP address range. It's not used if a single IP address is being specified. Only IPv4 address format is supported.

### EXAMPLES

```
Invoke-TSMCLI -CommandName delete-firewalliprule -ComputerName myserver -  
Port 80 -Authentication basic -Credential $credential @  
{ip="100.100.100.1"}
```

```
Invoke-TSMCLI -CommandName delete-firewalliprule -ComputerName myserver -  
Port 80 -Authentication basic -Credential $credential @  
{ipStart="100.100.100.1"; ipEnd="100.100.100.128"}
```

## RESPONSE

Object		Result
Integer	<b>Result</b>	The response code for "Success" (always 0).
String	<b>Message</b>	The localized message for "Success".

## delete-firewallportrule

Deletes a rule for blocking access to a single port or a port range of a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] delete-firewallportrule -ComputerName  
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]  
-Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-  
CommandArgs] @{protocol="<tcp|udp>"; port="<port number>"} [-LogLevel <0|  
1|2|3|4|5|6|7>]
```

```
Invoke-TSMCLI [-CommandName] delete-firewallportrule -ComputerName  
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]  
-Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-  
CommandArgs] @{protocol="<tcp|udp>"; portStart="<port number>";  
portEnd="<port number>"} [-LogLevel <0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The delete-firewallportrule command deletes a rule for blocking access to a single port or a port range of a specific Lenovo ThinkServer System Manager.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### protocol

The protocol type. It can be "tcp" or "udp".

#### port

Specifies a single port number. It's not used if a port range is being specified. Valid values are 1 to 65535.

#### portStart

Specifies a port range start. It's not used if a single port is being specified. Valid values are 1 to 65534.

#### portEnd

Specifies a port range end. It's not used if a single port is being specified. Valid values are 2 to 65535.



## EXAMPLES

```
Invoke-TSMCLI -CommandName delete-firewallportrule -ComputerName myserver
-Port 80 -Authentication basic -Credential $cred -CommandArgs @
{protocol="tcp"; port="50000"}
```

```
Invoke-TSMCLI -CommandName delete-firewallportrule -ComputerName myserver
-Port 80 -Authentication basic -Credential $cred -CommandArgs @
{protocol="udp"; portStart="50000"; portEnd="50100"}
```

## RESPONSE

Object Result		
Integer	<b>Result</b>	The response code for "Success" (always 0).
String	<b>Message</b>	The localized message for "Success".

## delete-landestination

Deletes a LAN Destination on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] delete-landestination -ComputerName
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]
-Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-
CommandArgs] @{lanDestinationId="<lan_destination_id>"} [-LogLevel <0|1|2|
3|4|5|6|7>]
```

### DESCRIPTION

The delete-landestination command deletes a LAN Destination on the event filters registered on a specific Lenovo ThinkServer System Manager.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### lanDestinationId

The id of the LAN Destination to be deleted.

### EXAMPLES

```
Invoke-TSMCLI -CommandName delete-landestination -ComputerName myserver -
Port 80 -Authentication basic -Credential $cred -CommandArgs @
{lanDestinationId="2"}
```

### RESPONSE

Object Result		
Integer	<b>Result</b>	The response code for "Success" (always 0).
String	<b>Message</b>	The localized message for "Success".

## delete-ldaprolegroup

Deletes an LDAP role group from a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] delete-ldaprolegroup -ComputerName
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]
-Credential <credential> [-UseSSL [-SkipSSLCertificateCheck]] [-
CommandArgs] @{roleGroupId=<role_group_id>} [-LogLevel <0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The delete-ldaprolegroup command deletes an LDAP role group from a specific Lenovo ThinkServer System Manager.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### roleGroupId

Specifies the role group identifier. To get the list of valid identifiers on the TSM, use the command "get-ldaprolegroups".

### EXAMPLES

```
Invoke-TSMCLI -CommandName delete-ldaprolegroup -ComputerName myserver -
Port 80 -Authentication basic -Credential $credential -CommandArgs @
{roleGroupId=1}
```

### RESPONSE

Object Result		
Integer	<b>Result</b>	The response code for "Success" (always 0).
String	<b>Message</b>	The localized message for "Success".

## delete-user

Deletes an user account from a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] delete-user -ComputerName <ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>] -Credential <credential> [-UseSSL [-SkipSSLCertificateCheck]] [-CommandArgs] @ {userId=<user_id>} [-LogLevel <0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The delete-user command deletes an user account from a specific Lenovo ThinkServer System Manager.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### userId

Specifies the user identifier. To get the list of valid user identifiers on the TSM, use the command "get-users".

### EXAMPLES

```
Invoke-TSMCLI -CommandName delete-user -ComputerName myserver -Port 80 -Authentication basic -Credential $credential -CommandArgs @{userId=4}
```

### RESPONSE

Object Result		
Integer	<b>Result</b>	The response code for "Success" (always 0).
String	<b>Message</b>	The localized message for "Success".

## enter-fwupdatemode

Enters the firmware update mode on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] enter-fwupdatemode -ComputerName
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]
-Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-LogLevel
<0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The enter-fwupdatemode command enters the firmware update mode on a specific Lenovo ThinkServer System Manager, enabling firmware update related commands. This command returns an identifier that needs to be informed when executing other firmware update commands.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### EXAMPLES

```
Invoke-TSMCLI -CommandName enter-fwupdatemode -ComputerName myserver -Port
80 -Authentication basic -Credential $cred
```

### RESPONSE

		Object Result
Integer	<b>Result</b>	The response code for "Success" (always 0).
String	<b>Message</b>	The localized message for "Success".
Long	<b>UpdateId</b>	The update identifier that needs to be used when exiting firmware update mode.

## exit-fwupdatemode

Exits the firmware update mode on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] exit-fwupdatemode -ComputerName
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]
-Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-
CommandArgs] @{updateId="<update_id>"} [-LogLevel <0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The exit-fwupdatemode command exits the firmware update mode on a specific Lenovo ThinkServer System Manager, resuming it to normal operation.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### updateId

Specifies the update mode identifier (required).

### EXAMPLES

```
Invoke-TSMCLI -CommandName exit-fwupdatemode -ComputerName myserver -Port
80 -Authentication basic -Credential $cred @{updateId="3199709292"}
```

### RESPONSE

Object Result		
Integer	<b>Result</b>	The response code for "Success" (always 0).
String	<b>Message</b>	The localized message for "Success".

## firmware-update

Performs a firmware update on a specific ThinkServer.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] firmware-update -ComputerName <ComputerName>
[-Port <PortNumber>] [-Authentication <AuthenticationMode>] -Credential
<Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-CommandArgs] @
{transferOrUploadFile="transfer|upload"[;fwFile="<path_to_image_file>"]
[;serverAddress="<server_ip_address>";shareType="<nfs|cifs|
tftp>";filename="<filename>"[;sourcePath="<source_path>";username="<userna
me>";password="<password>"]];rebootAfterUpdate="auto|manual"} [-LogLevel
<0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The firmware-update command is a bulk command that execute all the necessary steps to perform a firmware update on a ThinkServer. This command enter in firmware update mode, transfer or upload a firmware bundle to the TSM, start the firmware update and exit from firmware update mode.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### transferOrUploadFile

Defines how the bundle file will be send to the TSM to be used to update the ThinkServer components (required). Possible values are: "transfer" or "upload".

#### fwFile

The firmware image file to upload to the TSM (only required when transferOrUploadFile is "upload").

#### serverAddress

The remote media server address. It can be an IPv4 address or an IPv6 address (only required when transferOrUploadFile is "transfer").

#### shareType

The share type used on the specified server (only required when transferOrUploadFile is "transfer"). Valid values are "nfs", "cifs" or "tftp".

#### filename

The name of the image file that will be transferred to the TSM (only required when transferOrUploadFile is "transfer").

**sourcePath**

Path where the image files are located on the specified server (only required when transferOrUploadFile is "transfer" and shareType is "nfs" or "cifs").

**username**

The username for the server (only required when transferOrUploadFile is "transfer" and shareType is "cifs"). It shall have a maximum of 16 characters, must start with an alphabetical character and is case sensitive. The following special characters are not allowed: \[\]:;|=,+\*? <>@#%".

**password**

The user's password for the server (only required when transferOrUploadFile is "transfer" and shareType is "cifs"). All ASCII characters are accepted, except space.

**rebootAfterUpdate**

Defines the behavior for automatic reboots for some firmware update operations (required). Possible values are: "auto" or "manual". The update process for some components involves a TSM or Host reboot. In some of these cases, the reboot is needed during the update operation, and cannot be avoided. In some other cases, however, even though the update will take place only after rebooting, the reboot can be deferred by specifying the "manual" option, and will not happen automatically. Later on, when a reboot happens, the update is automatically applied. To automatically start the reboot, specify the "auto" option.

**EXAMPLES**

```
Invoke-TSMCLI -CommandName firmware-update -ComputerName myserver -Port 80
-Authentication basic -Credential $cred -CommandArgs @
{transferOrUploadFile="upload";fwFile="C:\MyFirmwareImageFolder
\myFirmwareFile";rebootAfterUpdate="auto"}
```

```
Invoke-TSMCLI -CommandName firmware-update -ComputerName myserver -Port 80
-Authentication basic -Credential $cred -CommandArgs @
{transferOrUploadFile="upload";fwFile="C:\MyFirmwareImageFolder
\myFirmwareFile";rebootAfterUpdate="manual"}
```

```
Invoke-TSMCLI -CommandName firmware-update -ComputerName myserver -Port 80
-Authentication basic -Credential $cred -CommandArgs @
{transferOrUploadFile="transfer";serverAddress="192.168.1.104";sourcePath=
"firmwareImages";shareType="cifs";username="myuser";password="1234";filena
me="myFirmwareFile";rebootAfterUpdate="auto"}
```

```
Invoke-TSMCLI -CommandName firmware-update -ComputerName myserver -Port 80
-Authentication basic -Credential $cred -CommandArgs @
{transferOrUploadFile="transfer";serverAddress="192.168.1.104";sourcePath=
"firmwareImages";shareType="cifs";username="myuser";password="1234";filena
me="myFirmwareFile";rebootAfterUpdate="manual"}
```

```
Invoke-TSMCLI -CommandName firmware-update -ComputerName myserver -Port 80
-Authentication basic -Credential $cred -CommandArgs @
{transferOrUploadFile="transfer";serverAddress="192.168.1.104";sourcePath=
"\firmwareImages";shareType="nfs";filename="myFirmwareFile";rebootAfterUpd
ate="auto"}
```



```
Invoke-TSMCLI -CommandName firmware-update -ComputerName myserver -Port 80
-Authentication basic -Credential $cred -CommandArgs @
{transferOrUploadFile="transfer";serverAddress="192.168.1.104";sourcePath=
"\firmwareImages";shareType="nfs";filename="myFirmwareFile";rebootAfterUpd
ate="manual"}
```

```
Invoke-TSMCLI -CommandName firmware-update -ComputerName myserver -Port 80
-Authentication basic -Credential $cred -CommandArgs @
{transferOrUploadFile="transfer";serverAddress="192.168.1.104";shareType="
tftp";filename="myFirmwareFile";rebootAfterUpdate="auto"}
```

```
Invoke-TSMCLI -CommandName firmware-update -ComputerName myserver -Port 80
-Authentication basic -Credential $cred -CommandArgs @
{transferOrUploadFile="transfer";serverAddress="192.168.1.104";shareType="
tftp";filename="myFirmwareFile";rebootAfterUpdate="manual"}
```

## RESPONSE

Object Result		
Integer	<b>Result</b>	The response code for "Success" (always 0).
String	<b>Message</b>	The localized message for "Success".

## get-adrolegroups

Gets the list of Active Directory role groups on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] get-adrolegroups -ComputerName <ComputerName>
[-Port <PortNumber>] [-Authentication <AuthenticationMode>] -Credential
<Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-LogLevel <0|1|2|3|4|
5|6|7>]
```

### DESCRIPTION

The get-adrolegroups command gets the list of Active Directory role groups on a specific Lenovo ThinkServer System Manager. Each role group maps an Active Directory group to a TSM role, and is used to control which access level is granted to this group.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### EXAMPLES

```
Invoke-TSMCLI -CommandName get-adrolegroups -ComputerName myserver -Port
80 -Authentication basic -Credential $credential
```

### RESPONSE

#### Array RoleGroups

Object RoleGroup								
Integer	<b>RoleGroupId</b>	The ID of the Active Directory role group.						
String	<b>GroupName</b>	The name of the Active Directory group.						
String	<b>DomainName</b>	The domain of the Active Directory group.						
Object	<b>Role</b>	<table border="1"> <tbody> <tr> <td>Integer</td> <td><b>Code</b></td> <td>The numeric code for the group role.</td> </tr> <tr> <td>String</td> <td><b>Description</b></td> <td>The localized description of the role.</td> </tr> </tbody> </table>	Integer	<b>Code</b>	The numeric code for the group role.	String	<b>Description</b>	The localized description of the role.
Integer	<b>Code</b>	The numeric code for the group role.						
String	<b>Description</b>	The localized description of the role.						
Boolean	<b>KVM</b>	"True" if the group has KVM privilege, "False" otherwise.						
Boolean	<b>VirtualMedia</b>	"True" if the group has Virtual Media privilege, "False" otherwise.						

Object RoleGroup

.  
. .  
.

**Object RoleGroup**

## get-adsettings

Gets the current Active Directory settings of a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] get-adsettings -ComputerName <ComputerName>
[-Port <PortNumber>] [-Authentication <AuthenticationMode>] -Credential
<credential> [-UseSSL [-SkipSSLCertificateCheck]] [-LogLevel <0|1|2|3|4|5|
6|7>]
```

### DESCRIPTION

The get-adsettings command gets the Active Directory information of a specific Lenovo ThinkServer System Manager, composed of a status, secret user, domain name and domain servers.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### EXAMPLES

```
Invoke-TSMCLI -CommandName get-adsettings -ComputerName myserver -Port 80
-Authentication basic -Credential $credential
```

### RESPONSE

Object ADSettings								
Object	<b>Status</b>	<table border="1"> <tr> <td>Integer</td> <td><b>Code</b></td> <td>The current code status for the Active Directory authentication.</td> </tr> <tr> <td>String</td> <td><b>Status</b></td> <td>The localized string representing the current Active Directory authentication status.</td> </tr> </table>	Integer	<b>Code</b>	The current code status for the Active Directory authentication.	String	<b>Status</b>	The localized string representing the current Active Directory authentication status.
Integer	<b>Code</b>	The current code status for the Active Directory authentication.						
String	<b>Status</b>	The localized string representing the current Active Directory authentication status.						
String	<b>SecretUser</b>	The username currently used to authenticate the Domain Controller.						
String	<b>DomainName</b>	The Domain Name of the Active Directory.						
String	<b>DomainServer1</b>	The first Domain Server IP address.						
String	<b>DomainServer2</b>	The second Domain Server IP address.						
String	<b>DomainServer3</b>	The third Domain Server IP address.						

## get-alertpolicies

Gets the list of Alert Policies on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] get-alertpolicies -ComputerName
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]
-Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-LogLevel
<0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The get-alertpolicies command gets the list of alert policies on a specific Lenovo ThinkServer System Manager.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### EXAMPLES

```
Invoke-TSMCLI -CommandName get-alertpolicies -ComputerName myserver -Port
80 -Authentication basic -Credential $cred
```

### RESPONSE

#### Array AlertPolicies

Object AlertPolicy								
Integer	<b>AlertPolicyId</b>	The ID of the Alert Policy.						
Integer	<b>AlertPolicyNumber</b>	The Alert Policy Number for the entry.						
Boolean	<b>AlertPolicyEnabled</b>	"True" if the Alert Policy is enabled, "False" otherwise.						
Object	<b>AlertPolicySet</b>	<table border="1"> <tbody> <tr> <td>Integer</td> <td><b>Code</b></td> <td>The numeric code for the Alert Policy Set.</td> </tr> <tr> <td>String</td> <td><b>Description</b></td> <td>The localized description for the Alert Policy Set.</td> </tr> </tbody> </table>	Integer	<b>Code</b>	The numeric code for the Alert Policy Set.	String	<b>Description</b>	The localized description for the Alert Policy Set.
Integer	<b>Code</b>	The numeric code for the Alert Policy Set.						
String	<b>Description</b>	The localized description for the Alert Policy Set.						
Integer	<b>LANDestinationId</b>	The LAN Destination ID to where the Alert Policy will redirect alerts.						
Boolean	<b>EventSpecificAlertString</b>	"True" if the alert will send an alert specific event string, "False" otherwise.						
Integer	<b>AlertStringKey</b>	The numeric key for the alert message string.						

Object AlertPolicy

.  
. .  
.

**Object AlertPolicy**

## get-auditlog

Gets the Audit Log from a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] get-auditlog -ComputerName <ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>] -Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-CommandArgs] @ {destinationPath="<path>"} [-LogLevel <0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The get-auditlog command gets the audit log from a specific Lenovo ThinkServer System Manager. The logs are saved on the specified path in a text file named using the prefix "AuditLog", the host name and a time stamp.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### destinationPath

Path to the folder where the log file will be saved.

### EXAMPLES

```
Invoke-TSMCLI -CommandName get-auditlog -ComputerName myserver -Port 80 -Authentication basic -Credential $cred -CommandArgs @ {destinationPath="C:\logs"}
```

### RESPONSE

Object Result		
Integer	<b>Result</b>	The response code for "Success" (always 0).
String	<b>Message</b>	The localized message for "Success".

## get-auditlogsettings

Gets the Audit Log settings on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] get-auditlogsettings -ComputerName
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]
-Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-LogLevel
<0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The get-auditlogsettings command gets if the Audit Log is enabled or not on a specific Lenovo ThinkServer System Manager.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### EXAMPLES

```
Invoke-TSMCLI -CommandName get-auditlogsettings -ComputerName myserver -
Port 80 -Authentication basic -Credential $cred
```

### RESPONSE

Object AuditLogSettings	
Boolean	<b>AuditLogEnabled</b> "True" if the Audit Log is enabled, "False" otherwise.



## get-commands

Lists all supported commands.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] get-commands [-LogLevel <0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The get-commands provides a list of all supported commands with a brief description.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### EXAMPLES

```
Invoke-TSMCLI -CommandName get-commands
```

## get-crashscreen

Gets crash screen image to a local file.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] get-crashscreen -ComputerName <ComputerName>
[-Authentication <AuthenticationMode>] -Credential <Credentials> [-
CommandArgs] @{{destinationPath="<image_file_path>"} [-LogLevel <0|1|2|3|4|
5|6|7>]}
```

### DESCRIPTION

The get-crashscreen command gets blue screen of death (BSOD) of host operating system from a specific Lenovo ThinkServer System Manager and save to a local file.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### destinationPath

Specifies the filename and path where the crash screen image will be saved (required).

### EXAMPLES

```
Invoke-TSMCLI -CommandName get-crashscreen -ComputerName myserver -
Authentication basic -Credential $cred -CommandArgs @{{destinationPath="C:
\crashscreen.jpeg"}}
```

### RESPONSE

Object Result		
Integer	<b>Code</b>	The response code for "Success" (always 0).
String	<b>Message</b>	The localized message for "Success".

## get-datetime

Gets the current date and time information of a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] get-datetime -ComputerName <ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>] -Credential <credential> [-UseSSL [-SkipSSLCertificateCheck]] [-LogLevel <0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The get-datetime command gets the date and time information of a specific Lenovo ThinkServer System Manager, composed of a timestamp in ISO 8601 format and a timezone identifier.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### EXAMPLES

```
Invoke-TSMCLI -CommandName get-datetime -ComputerName myserver -Port 80 -Authentication basic -Credential $credential
```

### RESPONSE

Object Result		
String	<b>DateTime</b>	The date and time string on a ISO8601 format.
String	<b>Timezone</b>	The string representing the current timezone.
DateTime	<b>DateTimeObject</b>	The date and time as a PowerShell DateTime object.

## get-dns

Gets the list of DNS servers configured on the TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] get-dns -ComputerName <ComputerName> [-Port
<PortNumber>] [-Authentication <AuthenticationMode>] -Credential
<credential> [-UseSSL [-SkipSSLCertificateCheck]] [-LogLevel <0|1|2|3|4|5|
6|7>]
```

### DESCRIPTION

The get-dns command gets the list of the DNS servers configured on a specific Lenovo ThinkServer System Manager. The list is ordered by the DNS server priority.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### EXAMPLES

```
Invoke-TSMCLI -CommandName get-dns -ComputerName myserver -Port 80 -
Authentication basic -Credential $credential
```

### RESPONSE

#### Array DNSServerAddresses

<p><b>String DNSServerAddress</b> The DNS server address.</p>
---

<p><b>String DNSServerAddress</b></p>
---------------------------------------

·  
·  
·

<p><b>String DNSServerAddress</b></p>
---------------------------------------

## get-eventfilters

Gets the list of platform event filters on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] get-eventfilters -ComputerName <ComputerName>
[-Port <PortNumber>] [-Authentication <AuthenticationMode>] -Credential
<Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-CommandArgs] @
{[eventFilterId="<event_filter_id>"]} [-LogLevel <0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The get-eventfilters command gets the list of platform event filters on a specific Lenovo ThinkServer System Manager.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### eventFilterId

Optional argument that specifies the id of the event filter to get information from. If this argument is not specified, information about all event filters is returned.

### EXAMPLES

```
Invoke-TSMCLI -CommandName get-eventfilters -ComputerName myserver -Port
80 -Authentication basic -Credential $cred -CommandArgs @
{eventFilterId="1"}
```

```
Invoke-TSMCLI -CommandName get-eventfilters -ComputerName myserver -Port
80 -Authentication basic -Credential $cred
```

### RESPONSE

#### Array EventFilters

		Object EventFilter
Integer	<b>EventFilterId</b>	The ID of the Event Filter.
Boolean	<b>EventFilterEnabled</b>	"True" if the event filter is enabled. "False" otherwise.
String	<b>Severity</b>	One of the following values: "unspecified", "monitor", "information", "normal", "non-critical", "critical" or "non-recoverable".

<p>Object <b>Action</b></p>	<table border="1"> <tr> <td data-bbox="558 248 821 309">String <b>PowerAction</b></td> <td data-bbox="821 248 1406 309">One of the following power actions: "none", "power-down", "power-reset" or "power-cycle".</td> </tr> <tr> <td data-bbox="558 320 821 358">Integer <b>AlertPolicyNumber</b></td> <td data-bbox="821 320 1406 358">The alert policy related to the Event Filter.</td> </tr> </table>	String <b>PowerAction</b>	One of the following power actions: "none", "power-down", "power-reset" or "power-cycle".	Integer <b>AlertPolicyNumber</b>	The alert policy related to the Event Filter.												
String <b>PowerAction</b>	One of the following power actions: "none", "power-down", "power-reset" or "power-cycle".																
Integer <b>AlertPolicyNumber</b>	The alert policy related to the Event Filter.																
<p>Object <b>GeneratorId</b></p>	<table border="1"> <tr> <td data-bbox="558 450 766 488">String <b>GeneratorId1</b></td> <td data-bbox="766 450 1037 488">The event filter generator ID 1.</td> </tr> <tr> <td data-bbox="558 499 766 537">String <b>GeneratorId2</b></td> <td data-bbox="766 499 1037 537">The event filter generator ID 2.</td> </tr> </table>	String <b>GeneratorId1</b>	The event filter generator ID 1.	String <b>GeneratorId2</b>	The event filter generator ID 2.												
String <b>GeneratorId1</b>	The event filter generator ID 1.																
String <b>GeneratorId2</b>	The event filter generator ID 2.																
<p>Object <b>Sensor</b></p>	<table border="1"> <tr> <td data-bbox="558 622 853 831"> <p>Object <b>SensorType</b></p> </td> <td data-bbox="853 622 1406 831"> <table border="1"> <tr> <td data-bbox="861 667 997 728">Integer <b>Code</b></td> <td data-bbox="997 667 1398 728">The numeric code of the sensor type related to the event filter.</td> </tr> <tr> <td data-bbox="861 739 997 799">String <b>Type</b></td> <td data-bbox="997 739 1398 799">The localized name of the sensor type related to the event filter.</td> </tr> </table> </td> </tr> <tr> <td data-bbox="558 842 821 880">String <b>SensorName</b></td> <td data-bbox="821 842 1406 880">The name of the sensor related to the event filter.</td> </tr> <tr> <td data-bbox="558 891 821 952">String <b>EventOptions</b></td> <td data-bbox="821 891 1406 952">Either "all" for all events or "sensor-events" for sensor specific events only.</td> </tr> <tr> <td data-bbox="558 963 853 1064"> <p>Array <b>SensorEvents</b> (Only available when <b>EventOptions</b> is "sensor-events")</p> </td> <td data-bbox="853 963 1406 1541"> <table border="1"> <tr> <td data-bbox="861 996 1398 1249"> <p><b>String Threshold</b></p> <p>One of the following available thresholds: "lower-non-critical-going-low", "lower-non-critical-going-high", "lower-critical-going-low", "lower-critical-going-high", "lower-non-recoverable-going-low", "lower-non-recoverable-going-high", "upper-non-critical-going-low", "upper-non-critical-going-high", "upper-critical-going-low", "upper-critical-going-high", "upper-non-recoverable-going-low", "upper-non-recoverable-going-high".</p> </td> </tr> <tr> <td data-bbox="861 1283 1070 1321"> <p><b>String Threshold</b></p> </td> </tr> <tr> <td data-bbox="1173 1373 1189 1429"> <p>⋮</p> </td> </tr> <tr> <td data-bbox="861 1462 1070 1500"> <p><b>String Threshold</b></p> </td> </tr> </table> </td> </tr> </table>	<p>Object <b>SensorType</b></p>	<table border="1"> <tr> <td data-bbox="861 667 997 728">Integer <b>Code</b></td> <td data-bbox="997 667 1398 728">The numeric code of the sensor type related to the event filter.</td> </tr> <tr> <td data-bbox="861 739 997 799">String <b>Type</b></td> <td data-bbox="997 739 1398 799">The localized name of the sensor type related to the event filter.</td> </tr> </table>	Integer <b>Code</b>	The numeric code of the sensor type related to the event filter.	String <b>Type</b>	The localized name of the sensor type related to the event filter.	String <b>SensorName</b>	The name of the sensor related to the event filter.	String <b>EventOptions</b>	Either "all" for all events or "sensor-events" for sensor specific events only.	<p>Array <b>SensorEvents</b> (Only available when <b>EventOptions</b> is "sensor-events")</p>	<table border="1"> <tr> <td data-bbox="861 996 1398 1249"> <p><b>String Threshold</b></p> <p>One of the following available thresholds: "lower-non-critical-going-low", "lower-non-critical-going-high", "lower-critical-going-low", "lower-critical-going-high", "lower-non-recoverable-going-low", "lower-non-recoverable-going-high", "upper-non-critical-going-low", "upper-non-critical-going-high", "upper-critical-going-low", "upper-critical-going-high", "upper-non-recoverable-going-low", "upper-non-recoverable-going-high".</p> </td> </tr> <tr> <td data-bbox="861 1283 1070 1321"> <p><b>String Threshold</b></p> </td> </tr> <tr> <td data-bbox="1173 1373 1189 1429"> <p>⋮</p> </td> </tr> <tr> <td data-bbox="861 1462 1070 1500"> <p><b>String Threshold</b></p> </td> </tr> </table>	<p><b>String Threshold</b></p> <p>One of the following available thresholds: "lower-non-critical-going-low", "lower-non-critical-going-high", "lower-critical-going-low", "lower-critical-going-high", "lower-non-recoverable-going-low", "lower-non-recoverable-going-high", "upper-non-critical-going-low", "upper-non-critical-going-high", "upper-critical-going-low", "upper-critical-going-high", "upper-non-recoverable-going-low", "upper-non-recoverable-going-high".</p>	<p><b>String Threshold</b></p>	<p>⋮</p>	<p><b>String Threshold</b></p>
<p>Object <b>SensorType</b></p>	<table border="1"> <tr> <td data-bbox="861 667 997 728">Integer <b>Code</b></td> <td data-bbox="997 667 1398 728">The numeric code of the sensor type related to the event filter.</td> </tr> <tr> <td data-bbox="861 739 997 799">String <b>Type</b></td> <td data-bbox="997 739 1398 799">The localized name of the sensor type related to the event filter.</td> </tr> </table>	Integer <b>Code</b>	The numeric code of the sensor type related to the event filter.	String <b>Type</b>	The localized name of the sensor type related to the event filter.												
Integer <b>Code</b>	The numeric code of the sensor type related to the event filter.																
String <b>Type</b>	The localized name of the sensor type related to the event filter.																
String <b>SensorName</b>	The name of the sensor related to the event filter.																
String <b>EventOptions</b>	Either "all" for all events or "sensor-events" for sensor specific events only.																
<p>Array <b>SensorEvents</b> (Only available when <b>EventOptions</b> is "sensor-events")</p>	<table border="1"> <tr> <td data-bbox="861 996 1398 1249"> <p><b>String Threshold</b></p> <p>One of the following available thresholds: "lower-non-critical-going-low", "lower-non-critical-going-high", "lower-critical-going-low", "lower-critical-going-high", "lower-non-recoverable-going-low", "lower-non-recoverable-going-high", "upper-non-critical-going-low", "upper-non-critical-going-high", "upper-critical-going-low", "upper-critical-going-high", "upper-non-recoverable-going-low", "upper-non-recoverable-going-high".</p> </td> </tr> <tr> <td data-bbox="861 1283 1070 1321"> <p><b>String Threshold</b></p> </td> </tr> <tr> <td data-bbox="1173 1373 1189 1429"> <p>⋮</p> </td> </tr> <tr> <td data-bbox="861 1462 1070 1500"> <p><b>String Threshold</b></p> </td> </tr> </table>	<p><b>String Threshold</b></p> <p>One of the following available thresholds: "lower-non-critical-going-low", "lower-non-critical-going-high", "lower-critical-going-low", "lower-critical-going-high", "lower-non-recoverable-going-low", "lower-non-recoverable-going-high", "upper-non-critical-going-low", "upper-non-critical-going-high", "upper-critical-going-low", "upper-critical-going-high", "upper-non-recoverable-going-low", "upper-non-recoverable-going-high".</p>	<p><b>String Threshold</b></p>	<p>⋮</p>	<p><b>String Threshold</b></p>												
<p><b>String Threshold</b></p> <p>One of the following available thresholds: "lower-non-critical-going-low", "lower-non-critical-going-high", "lower-critical-going-low", "lower-critical-going-high", "lower-non-recoverable-going-low", "lower-non-recoverable-going-high", "upper-non-critical-going-low", "upper-non-critical-going-high", "upper-critical-going-low", "upper-critical-going-high", "upper-non-recoverable-going-low", "upper-non-recoverable-going-high".</p>																	
<p><b>String Threshold</b></p>																	
<p>⋮</p>																	
<p><b>String Threshold</b></p>																	

Object	<b>EventData</b>	
Integer	<b>EventTrigger</b>	The event/reading type value.
Integer	<b>EventData1ANDMask</b>	The wildcarded or compared bits.
Integer	<b>EventData1Compare1</b>	This field is used to indicate whether each bit position's comparison is an exact comparison or not.
Integer	<b>EventData1Compare2</b>	This field is used to indicate whether each bit position's comparison is an exact comparison or not.
Integer	<b>EventData2ANDMask</b>	The wildcarded or compared bits.
Integer	<b>EventData2Compare1</b>	This field is used to indicate whether each bit position's comparison is an exact comparison or not.
Integer	<b>EventData2Compare2</b>	This field is used to indicate whether each bit position's comparison is an exact comparison or not.
Integer	<b>EventData3ANDMask</b>	The wildcarded or compared bits.
Integer	<b>EventData3Compare1</b>	This field is used to indicate whether each bit position's comparison is an exact comparison or not.
Integer	<b>EventData3Compare2</b>	This field is used to indicate whether each bit position's comparison is an exact comparison or not.

**Object EventFilter**

·  
·  
·

**Object EventFilter**

## get-firewallrules

Gets the firewall rules from a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] get-firewallrules -ComputerName
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]
-Credential <credential> [-UseSSL [-SkipSSLCertificateCheck]] [-LogLevel
<0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The get-firewallrules command gets the firewall rules from a specific Lenovo ThinkServer System Manager.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### EXAMPLES

```
Invoke-TSMCLI -CommandName get-firewallrules -ComputerName myserver -Port
80 -Authentication basic -Credential $credential
```

### RESPONSE

Object FirewallRules									
Array	<b>IPrules</b>								
	<table border="1"> <thead> <tr> <th colspan="2">Object IPRule</th> </tr> </thead> <tbody> <tr> <td>Integer</td> <td><b>Index</b> The index of the IP firewall rule on the list.</td> </tr> <tr> <td>String</td> <td><b>IPStart</b> The IP address being blocked or the starting IP on a range of addresses.</td> </tr> <tr> <td>String</td> <td><b>IPEnd</b> The ending IP on a range of addresses.</td> </tr> </tbody> </table>	Object IPRule		Integer	<b>Index</b> The index of the IP firewall rule on the list.	String	<b>IPStart</b> The IP address being blocked or the starting IP on a range of addresses.	String	<b>IPEnd</b> The ending IP on a range of addresses.
Object IPRule									
Integer	<b>Index</b> The index of the IP firewall rule on the list.								
String	<b>IPStart</b> The IP address being blocked or the starting IP on a range of addresses.								
String	<b>IPEnd</b> The ending IP on a range of addresses.								
	<table border="1"> <tr> <td><b>Object IPRule</b></td> </tr> <tr> <td style="text-align: center;">⋮</td> </tr> <tr> <td><b>Object IPRule</b></td> </tr> </table>	<b>Object IPRule</b>	⋮	<b>Object IPRule</b>					
<b>Object IPRule</b>									
⋮									
<b>Object IPRule</b>									



Array **PortRules**

Object PortRule		
Integer	<b>Index</b>	The index of the port firewall rule on the list.
String	<b>PortStart</b>	The port number being blocked or the starting port on a range of ports.
String	<b>PortEnd</b>	The ending port on a range of ports.
String	<b>Protocol</b>	The protocol being blocked by the rule (either "tcp" or "udp").

Object PortRule

⋮

Object PortRule

## get-frus

Gets the list of available Field Replaceable Units (FRUs) on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] get-frus -ComputerName <ComputerName> [-Port
<PortNumber>] [-Authentication <AuthenticationMode>] -Credential
<credential> [-UseSSL [-SkipSSLCertificateCheck]] [-LogLevel <0|1|2|3|4|5|
6|7>]
```

### DESCRIPTION

The get-frus command gets the list of available Field Replaceable Units (FRUs) on a specific Lenovo ThinkServer System Manager.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### EXAMPLES

```
Invoke-TSMCLI -CommandName get-frus -ComputerName myserver -Port 80 -
Authentication basic -Credential $credential
```

### RESPONSE

#### Array FRUInventory

Object FRU (Note that the fields described below may not be present on all FRU entries)								
Integer	<b>FRUDeviceID</b>	FRU device ID.						
String	<b>FRUDeviceName</b>	FRU device name.						
Integer	<b>ChassisInfoAreaFormatVersion</b>	Chassis information area format version.						
String	<b>ChassisType</b>	Chassis type.						
String	<b>ChassisPartNum</b>	Chassis part number.						
String	<b>ChassisSerialNum</b>	Chassis serial number.						
String	<b>ChassisExtra</b>	Chassis extra information.						
Integer	<b>BoardInfoAreaFormatVersion</b>	Board information area format version.						
Object	<b>BoardLanguage</b>	<table border="1"> <tbody> <tr> <td>Integer</td> <td><b>Code</b></td> <td>The numeric code for the language.</td> </tr> <tr> <td>String</td> <td><b>Language</b></td> <td>The language name.</td> </tr> </tbody> </table>	Integer	<b>Code</b>	The numeric code for the language.	String	<b>Language</b>	The language name.
Integer	<b>Code</b>	The numeric code for the language.						
String	<b>Language</b>	The language name.						
String	<b>BoardManufactureDateTime</b>	Board manufacture date and time.						

String	<b>BoardManufacturer</b>	Board manufacturer.						
String	<b>BoardProductName</b>	Board product name.						
String	<b>BoardSerialNum</b>	Board serial number.						
String	<b>BoardPartNum</b>	Board part number.						
String	<b>BoardFRUFileID</b>	Board FRU file ID.						
String	<b>BoardExtra</b>	Board extra information.						
Integer	<b>ProductInfoAreaFormatVersion</b>	Product information area format version.						
Object	<b>ProductLanguage</b>	<table border="1"> <tr> <td>Integer</td> <td><b>Code</b></td> <td>The numeric code for the language.</td> </tr> <tr> <td>String</td> <td><b>Language</b></td> <td>The language name.</td> </tr> </table>	Integer	<b>Code</b>	The numeric code for the language.	String	<b>Language</b>	The language name.
Integer	<b>Code</b>	The numeric code for the language.						
String	<b>Language</b>	The language name.						
String	<b>ProductManufacturerName</b>	Product manufacturer name.						
String	<b>ProductName</b>	Product name.						
String	<b>ProductDate</b>	Product date.						
String	<b>ProductPartNum</b>	Product part number.						
String	<b>ProductVersion</b>	Product version.						
String	<b>ProductSerialNum</b>	Product serial number.						
Integer	<b>SlotNumber</b>	Slot number.						
String	<b>ProductManufactureDate</b>	Product manufacture date.						
String	<b>ProductAssetTag</b>	Product asset tag.						
String	<b>ProductFRUFileID</b>	Product FRU file ID.						
String	<b>ProductExtra</b>	Product extra information.						
String	<b>MemorySlotIndex</b>	Memory slot index.						
Integer	<b>ChannelNum</b>	Channel number.						
String	<b>DIMMType</b>	DIMM type.						
String	<b>DDRVoltage</b>	DDR voltage.						
String	<b>MemoryFrequency</b>	Memory frequency.						
String	<b>DIMMSize</b>	DIMM size.						
Integer	<b>CPUIndex</b>	CPU index.						
Integer	<b>CoreCount</b>	Core count.						
Integer	<b>ThreadCount</b>	Thread count.						
String	<b>CPUVendor</b>	CPU vendor.						
String	<b>CPUFamily</b>	CPU family.						
String	<b>CPUModel</b>	CPU model.						
String	<b>Stepping</b>	Stepping.						
String	<b>MaxFrequency</b>	Maximum frequency.						
Integer	<b>ControllerIndex</b>	Controller index.						
Integer	<b>DriveIndex</b>	Drive index.						
String	<b>VendorID</b>	Vendor ID.						
String	<b>Size</b>	Drive size.						
String	<b>MediaType</b>	Media type.						
String	<b>InterfaceType</b>	Interface type.						

String	<b>FormFactor</b>	Form factor.
String	<b>LinkSpeed</b>	Link speed.
String	<b>DeviceState</b>	Device state.

**Object FRU (Note that the fields described below may not be present on all FRU entries)**

.

.

.

**Object FRU (Note that the fields described below may not be present on all FRU entries)**

## get-fwupdatestatus

Gets the firmware update status of a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] get-fwupdatestatus -ComputerName
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]
-Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-LogLevel
<0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The get-fwupdatestatus command gets the firmware update status of a specific Lenovo ThinkServer System Manager.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### EXAMPLES

```
Invoke-TSMCLI -CommandName get-fwupdatestatus -ComputerName myserver -Port
80 -Authentication basic -Credential $cred
```

### RESPONSE

#### Array FWUpdates

Object FWUpdate							
Object <b>DeviceType</b>	<table border="1"> <tr> <td>Integer</td> <td><b>Code</b></td> <td>The numeric code for the device type.</td> </tr> <tr> <td>String</td> <td><b>Type</b></td> <td>The device type.</td> </tr> </table>	Integer	<b>Code</b>	The numeric code for the device type.	String	<b>Type</b>	The device type.
Integer	<b>Code</b>	The numeric code for the device type.					
String	<b>Type</b>	The device type.					
Object <b>UpdateStatus</b>	<table border="1"> <tr> <td>Integer</td> <td><b>Code</b></td> <td>The update status numeric code.</td> </tr> <tr> <td>String</td> <td><b>Status</b></td> <td>A string describing the update status.</td> </tr> </table>	Integer	<b>Code</b>	The update status numeric code.	String	<b>Status</b>	A string describing the update status.
Integer	<b>Code</b>	The update status numeric code.					
String	<b>Status</b>	A string describing the update status.					
String <b>Version</b>	The image version.						
Integer <b>UpdatePercentage</b>	The completion percentage of the update.						

Object FWUpdate

.  
. .  
.

**Object FWUpdate**

## get-fwversions

Gets the current and the new (if available) firmware versions for components of a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] get-fwversions -ComputerName <ComputerName>
[-Port <PortNumber>] [-Authentication <AuthenticationMode>] -Credential
<Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-LogLevel <0|1|2|3|4|
5|6|7>]
```

### DESCRIPTION

The get-fwversions command gets the current and the new (if available) firmware versions of a specific Lenovo ThinkServer System Manager.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### EXAMPLES

```
Invoke-TSMCLI -CommandName get-fwversions -ComputerName myserver -Port 80
-Authentication basic -Credential $cred
```

### RESPONSE

#### Array FWImages

		Object FWImage						
Object	<b>DeviceType</b>	<table border="1"> <tr> <td>Integer</td> <td><b>Code</b></td> <td>The numeric code for the device type.</td> </tr> <tr> <td>String</td> <td><b>Type</b></td> <td>The device type.</td> </tr> </table>	Integer	<b>Code</b>	The numeric code for the device type.	String	<b>Type</b>	The device type.
Integer	<b>Code</b>	The numeric code for the device type.						
String	<b>Type</b>	The device type.						
Object	<b>DeviceStatus</b>	<table border="1"> <tr> <td>Integer</td> <td><b>Code</b></td> <td>The device status numeric code.</td> </tr> <tr> <td>String</td> <td><b>Status</b></td> <td>A string describing the device status.</td> </tr> </table>	Integer	<b>Code</b>	The device status numeric code.	String	<b>Status</b>	A string describing the device status.
Integer	<b>Code</b>	The device status numeric code.						
String	<b>Status</b>	A string describing the device status.						
Object	<b>StatusFlag</b>	<table border="1"> <tr> <td>Integer</td> <td><b>Code</b></td> <td>The status flag status numeric code.</td> </tr> <tr> <td>String</td> <td><b>StatusFlag</b></td> <td>A string describing the status flag.</td> </tr> </table>	Integer	<b>Code</b>	The status flag status numeric code.	String	<b>StatusFlag</b>	A string describing the status flag.
Integer	<b>Code</b>	The status flag status numeric code.						
String	<b>StatusFlag</b>	A string describing the status flag.						

Object <b>Reboot</b>	<table border="1"> <tr> <td>Integer</td> <td><b>Code</b></td> <td>A numeric code that describes if either the host or TSM needs or causes a reboot after updating this component.</td> </tr> <tr> <td>String</td> <td><b>Reboot</b></td> <td>A string that describes if either the host or TSM needs or causes a reboot after updating this component.</td> </tr> </table>	Integer	<b>Code</b>	A numeric code that describes if either the host or TSM needs or causes a reboot after updating this component.	String	<b>Reboot</b>	A string that describes if either the host or TSM needs or causes a reboot after updating this component.
Integer	<b>Code</b>	A numeric code that describes if either the host or TSM needs or causes a reboot after updating this component.					
String	<b>Reboot</b>	A string that describes if either the host or TSM needs or causes a reboot after updating this component.					
String	<b>CurrentImageVersion</b>	The currently installed image version.					
String	<b>NewImageVersion</b>	The version of the image transferred or uploaded to the TSM.					

**Object FWImage**

.  
.  
.

**Object FWImage**



## get-help

Provides detailed help information for a command.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] get-help [[-CommandArgs] @  
{commandName="<command>"}] [-LogLevel <0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The get-help command provides a help guide for supported commands. It describes the command functionality and its syntax.

To list all available commands use this command: Invoke-TSMCLI -CommandName get-commands

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### **commandName**

Specifies the command for which it is desired to obtain the help message. Examples: "create-credential", "get-powerstate".

### EXAMPLES

```
Invoke-TSMCLI -CommandName get-help -CommandArgs @{commandName="create-  
credential"}
```

## get-identfierstate

Provides state information about Chassis status Led.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] get-identfierstate -ComputerName
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]
-Credential <credential> [-UseSSL [-SkipSSLCertificateCheck] [-LogLevel
<0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The get-identfierstate command retrieves the physical system identification state, typically a LED.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### EXAMPLES

```
Invoke-TSMCLI -CommandName get-identfierstate -ComputerName myserver -
Port 80 -Authentication basic -Credential $credential
```

### RESPONSE

		Object Result
Integer	<b>State</b>	The physical system identification state. Off (0), On with timer (1), or On indefinitely (2).
String	<b>Description</b>	The description message of the system identification state.

## get-landestinations

Gets the list of LAN destinations on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] get-landestinations -ComputerName
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]
-Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-LogLevel
<0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The get-landestinations command gets the list of LAN destinations for the platform event filters (PEF) on a specific Lenovo ThinkServer System Manager.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### EXAMPLES

```
Invoke-TSMCLI -CommandName get-landestinations -ComputerName myserver -
Port 80 -Authentication basic -Credential $cred
```

### RESPONSE

#### Array LANDestinations

		Object LANDestination
Integer	<b>LANDestinationId</b>	The ID of the LAN destination entry.
String	<b>Type</b>	The type of LAN destination. Either "snmp" or "email".
String	<b>DestinationAddress</b>	The address for the SNMP trap. (Only when type is "snmp")
Integer	<b>UserId</b>	The ID of the user to whom the email message will be sent. (Only when type is "email")
String	<b>Username</b>	The username of the user to whom the email message will be sent. (Only when type is "email")

Object LANDestination

·  
·  
·

Object LANDestination

## get-ldaprolegroups

Gets the list of LDAP role groups on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] get-ldaprolegroups -ComputerName
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]
-Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-LogLevel
<0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The get-ldaprolegroups command gets the list of LDAP role groups on a specific Lenovo ThinkServer System Manager. Each role group maps an LDAP group to a TSM role, and is used to control which access level is granted to this group.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### EXAMPLES

```
Invoke-TSMCLI -CommandName get-ldaprolegroups -ComputerName myserver -Port
80 -Authentication basic -Credential $credential
```

### RESPONSE

#### Array RoleGroups

Object RoleGroup								
Integer	<b>RoleGroupId</b>	The ID of the LDAP role group.						
String	<b>GroupName</b>	The name of the LDAP group.						
String	<b>GroupSearchBase</b>	The search base of the LDAP group.						
Object	<b>Role</b>	<table border="1"> <tbody> <tr> <td>String</td> <td><b>Code</b></td> <td>The code for the group role.</td> </tr> <tr> <td>String</td> <td><b>Description</b></td> <td>The localized description of the role.</td> </tr> </tbody> </table>	String	<b>Code</b>	The code for the group role.	String	<b>Description</b>	The localized description of the role.
String	<b>Code</b>	The code for the group role.						
String	<b>Description</b>	The localized description of the role.						
Boolean	<b>KVM</b>	"True" if the group has KVM privilege, "False" otherwise.						
Boolean	<b>VirtualMedia</b>	"True" if the group has Virtual Media privilege, "False" otherwise.						

Object RoleGroup

.  
. .  
.

**Object RoleGroup**

## get-ldapsettings

Gets the current LDAP settings from a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] get-ldapsettings -ComputerName <ComputerName>
[-Port <PortNumber>] [-Authentication <AuthenticationMode>] -Credential
<credential> [-UseSSL [-SkipSSLCertificateCheck]] [-LogLevel <0|1|2|3|4|5|
6|7>]
```

### DESCRIPTION

The get-ldapsettings command gets the current LDAP settings from a specific Lenovo ThinkServer System Manager.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### EXAMPLES

```
Invoke-TSMCLI -CommandName get-ldapsettings -ComputerName myserver -Port
80 -Authentication basic -Credential $credential
```

### RESPONSE

Object LDAPSettings								
Object	<b>Status</b>	<table border="1"> <tr> <td>Integer</td> <td><b>Code</b></td> <td>The current code status for the LDAP authentication.</td> </tr> <tr> <td>String</td> <td><b>Status</b></td> <td>The localized string representing the current LDAP authentication status (Enabled/Disabled).</td> </tr> </table>	Integer	<b>Code</b>	The current code status for the LDAP authentication.	String	<b>Status</b>	The localized string representing the current LDAP authentication status (Enabled/Disabled).
Integer	<b>Code</b>	The current code status for the LDAP authentication.						
String	<b>Status</b>	The localized string representing the current LDAP authentication status (Enabled/Disabled).						
Object	<b>EncryptedType</b>	<table border="1"> <tr> <td>String</td> <td><b>Code</b></td> <td>The current code for the encryption type used on LDAP authentication.</td> </tr> <tr> <td>String</td> <td><b>Description</b></td> <td>The localized string representing the current encryption type used on LDAP authentication (No Encryption/SSL/StartTLS).</td> </tr> </table>	String	<b>Code</b>	The current code for the encryption type used on LDAP authentication.	String	<b>Description</b>	The localized string representing the current encryption type used on LDAP authentication (No Encryption/SSL/StartTLS).
String	<b>Code</b>	The current code for the encryption type used on LDAP authentication.						
String	<b>Description</b>	The localized string representing the current encryption type used on LDAP authentication (No Encryption/SSL/StartTLS).						
Object	<b>CommonNameType</b>	<table border="1"> <tr> <td>String</td> <td><b>Code</b></td> <td>The current code for the common name type of the LDAP server.</td> </tr> <tr> <td>String</td> <td><b>Description</b></td> <td>The localized string representing the common name type of the LDAP server (IP Address/FQDN).</td> </tr> </table>	String	<b>Code</b>	The current code for the common name type of the LDAP server.	String	<b>Description</b>	The localized string representing the common name type of the LDAP server (IP Address/FQDN).
String	<b>Code</b>	The current code for the common name type of the LDAP server.						
String	<b>Description</b>	The localized string representing the common name type of the LDAP server (IP Address/FQDN).						
String	<b>ServerAddress</b>	The IP address or the fully qualified domain name (FQDN) of the LDAP server.						
Integer	<b>Port</b>	The port number of the LDAP server.						

String	<b>BindDN</b>	The Bind DN configuration for the LDAP server.
String	<b>SearchBase</b>	The search base configuration for the LDAP server.
String	<b>AttributeOfUserLogin</b>	The attribute of user login tells the LDAP server which attribute should be used to identify the user (either "cn" or "uid").

## get-niclinksettings

Gets the network link settings on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] get-niclinksettings -ComputerName
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]
-Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-LogLevel
<0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The get-niclinksettings command gets the network link configuration from the currently active network interface on a specific Lenovo ThinkServer System Manager. This configuration is formed by NIC mode, link mode, link speed and duplex mode settings.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### EXAMPLES

```
Invoke-TSMCLI -CommandName get-niclinksettings -ComputerName myserver -
Port 80 -Authentication basic -Credential $cred
```

### RESPONSE

Object NICLinkSettings		
String	<b>NICMode</b>	"dedicated" if the dedicated NIC is enabled or "shared" if the shared NIC is enabled.
String	<b>LinkMode</b>	"auto" if the the NIC link auto-negotiation is enabled, "manual" if the NIC link is manually configured.
Integer	<b>LinkSpeed</b>	The NIC link speed in Mbps.
Boolean	<b>FullDuplex</b>	"TRUE" if the NIC is operating in full-duplex, "FALSE" if the NIC is operating in half-duplex.



## get-nics

Lists network interfaces of a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] get-nics -ComputerName <ComputerName> [-Port
<PortNumber>] [-Authentication <AuthenticationMode>] -Credential
<credential> [-UseSSL [-SkipSSLCertificateCheck]] [-LogLevel <0|1|2|3|4|5|
6|7>]
```

### DESCRIPTION

The get-nics command lists network interfaces of a specific Lenovo ThinkServer System Manager. It returns the identifier and description of each network interface.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### EXAMPLES

```
Invoke-TSMCLI -CommandName get-nics -ComputerName myserver -Port 80 -
Authentication basic -Credential $credential
```

### RESPONSE

#### Array NICs

Object NIC	
String	<b>NicId</b> The ID of the network interface.
Object	<b>NICMode</b>
String	<b>Code</b> The network interface mode either "shared" or "dedicated".
String	<b>Description</b> The localized description for the NICMode code.

Object NIC

·  
·  
·

Object NIC

## get-nicsettings

Gets the network configuration of the active network interface on the TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] get-nicsettings -ComputerName <ComputerName>
[-Port <PortNumber>] [-Authentication <AuthenticationMode>] -Credential
<Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-LogLevel <0|1|2|3|4|
5|6|7>]
```

### DESCRIPTION

The get-nicsettings command gets the network configuration of a specific Lenovo ThinkServer System Manager.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### EXAMPLES

```
Invoke-TSMCLI -CommandName get-nicsettings -ComputerName myserver -Port 80
-Authentication basic -Credential $cred
```

### RESPONSE

Object NICSettings														
String	<b>NICId</b>	The LAN Interface name.												
Boolean	<b>NICEnabled</b>	Enabled or Disable LAN settings for the LAN interface.												
Object	<b>NICMode</b>	<table border="1"> <tr> <td>String</td> <td><b>Code</b></td> <td>The network interface mode either "shared" or "dedicated".</td> </tr> <tr> <td>String</td> <td><b>Description</b></td> <td>The localized description for the NICMode code.</td> </tr> </table>	String	<b>Code</b>	The network interface mode either "shared" or "dedicated".	String	<b>Description</b>	The localized description for the NICMode code.						
String	<b>Code</b>	The network interface mode either "shared" or "dedicated".												
String	<b>Description</b>	The localized description for the NICMode code.												
String	<b>MACAddress</b>	The MAC address for that NIC.												
Object	<b>IPv4</b>	<table border="1"> <tr> <td>String</td> <td><b>Mode</b></td> <td>The IP address assignment mode. Either "dhcp" or "static".</td> </tr> <tr> <td>String</td> <td><b>Address</b></td> <td>The IPv4 address.</td> </tr> <tr> <td>String</td> <td><b>SubnetMask</b></td> <td>The IPv4 address subnet mask.</td> </tr> <tr> <td>String</td> <td><b>DefaultGateway</b></td> <td>The IPv4 default gateway for the NIC.</td> </tr> </table>	String	<b>Mode</b>	The IP address assignment mode. Either "dhcp" or "static".	String	<b>Address</b>	The IPv4 address.	String	<b>SubnetMask</b>	The IPv4 address subnet mask.	String	<b>DefaultGateway</b>	The IPv4 default gateway for the NIC.
String	<b>Mode</b>	The IP address assignment mode. Either "dhcp" or "static".												
String	<b>Address</b>	The IPv4 address.												
String	<b>SubnetMask</b>	The IPv4 address subnet mask.												
String	<b>DefaultGateway</b>	The IPv4 default gateway for the NIC.												

Object	<b>IPv6</b>
Boolean	<b>Enabled</b> True if IPv6 is enabled for the NIC, False otherwise.
String	<b>Mode</b> The IP address assignment mode. Either "dhcp" or "static".
String	<b>Address</b> The IPv6 address.
String	<b>SubnetPrefixLength</b> The IPv6 address subnet prefix length.
String	<b>DefaultGateway</b> The IPv6 default gateway for the NIC.
Object	<b>VLAN</b>
Boolean	<b>Enabled</b> True if the VLAN is enabled, False otherwise.
Integer	<b>Id</b> The ID for the VLAN settings.
Integer	<b>Priority</b> The priority for the VLAN settings.

## get-ntpsettings

Gets the current network time server (NTP) configuration of a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] get-ntpsettings -ComputerName <ComputerName>
[-Port <PortNumber>] [-Authentication <AuthenticationMode>] -Credential
<credential> [-UseSSL [-SkipSSLCertificateCheck]] [-LogLevel <0|1|2|3|4|5|
6|7>]
```

### DESCRIPTION

The get-ntpsettings command gets the network time server (NTP) configuration of a specific Lenovo ThinkServer System Manager. The configuration contains the primary and secondary NTP servers and the synchronization status.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### EXAMPLES

```
Invoke-TSMCLI -CommandName get-ntpsettings -ComputerName myserver -Port 80
-Authentication basic -Credential $credential
```

### RESPONSE

Object NTPSettings							
String	<b>NTPServer1</b> The primary NTP server.						
String	<b>NTPServer2</b> The secondary NTP server.						
Object	<b>Status</b> <table border="1" data-bbox="480 1543 1145 1675"> <tbody> <tr> <td>Integer</td> <td><b>Code</b></td> <td>The numeric code for the NTP synchronization status. 1 when automatically synchronize date and time. 2 when unable to synchronize date and time.</td> </tr> <tr> <td>String</td> <td><b>Description</b></td> <td>The localized string for the NTP synchronization status.</td> </tr> </tbody> </table>	Integer	<b>Code</b>	The numeric code for the NTP synchronization status. 1 when automatically synchronize date and time. 2 when unable to synchronize date and time.	String	<b>Description</b>	The localized string for the NTP synchronization status.
Integer	<b>Code</b>	The numeric code for the NTP synchronization status. 1 when automatically synchronize date and time. 2 when unable to synchronize date and time.					
String	<b>Description</b>	The localized string for the NTP synchronization status.					

## get-pamorder

Gets the PAM order on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] get-pamorder -ComputerName <ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>] -Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-LogLevel <0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The get-pamorder command gets the ordered list of Pluggable Authentication Module (PAM) on a specific Lenovo ThinkServer System Manager.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### EXAMPLES

```
Invoke-TSMCLI -CommandName get-pamorder -ComputerName myserver -Port 80 -Authentication basic -Credential $cred
```

### RESPONSE

#### Array PAMOrder

<b>String PAMModule</b>
One of the following PAM modules: "ipmi" for IPMI, "ldap" for LDAP or "ad" for Active Directory.

**String PAMModule**

·  
·  
·

**String PAMModule**

## get-powerbuttonstate

Gets the current power button state ("true" if enabled or "false" if disabled) from a specific ThinkServer.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] get-powerbuttonstate -ComputerName  
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]  
-Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-LogLevel  
<0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The get-powerbuttonstate command gets the current power button state ("true" if enabled or "false" if disabled) from a specific ThinkServer.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### EXAMPLES

```
Invoke-TSMCLI -CommandName get-powerbuttonstate -ComputerName myserver -  
Port 80 -Authentication basic -Credential $cred
```

### RESPONSE

Boolean PowerButtonEnabled
----------------------------

## get-powerconsumption

Retrieves information about a ThinkServer power consumption. It provides the current power consumption, the average power consumption (based on values recorded by TSM), the date and time of the most recent peak of consumption and also the power consumption history.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] get-powerconsumption -ComputerName
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]
-Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-
CommandArgs] [ @{getHistory=<0|1>;saveToFile="<filepath>" } ] [-LogLevel
<0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The `get-powerconsumption` command retrieves the information about power consumption (in watts) from a specific ThinkServer. It provides the current power consumption, the average power consumption (based on values recorded by TSM), the date and time of the most recent peak of consumption and also the power consumption history.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### getHistory

Indicates if the user desires to get the history of all power consumption information obtained from the ThinkServer (optional). Valid values: 0 (default) for no history or 1 to get the history.

#### saveToFile

Allows the definition of filename and path where the power consumption history of the ThinkServer will be saved (applicable only if `getHistory` is "1"). It will be saved in a CSV format with index, date and time (in ISO 8601 format) and the power consumption value. The path shall exist, and if there is already a file with the same name on that path, it will be overwritten. The filename must have an extension.

### EXAMPLES

```
Invoke-TSMCLI get-powerconsumption -ComputerName myserver -Port 80 -
Authentication basic -Credential $cred
```

```
Invoke-TSMCLI get-powerconsumption -ComputerName myserver -Port 80 -
Authentication basic -Credential $cred @{getHistory="1"}
```

```
Invoke-TSMCLI get-powerconsumption -ComputerName myserver -Port 80 -
Authentication basic -Credential $cred @
{getHistory="1";saveToFile="powerhistory.csv"}
```

## RESPONSE

<b>Object Result (Note that some fields described below may not be present)</b>	
String	<b>Current</b> The current power consumption (in watts).
String	<b>Average</b> The average power consumption for the recorded period (in watts).
Object	<b>Peak</b>
DateTime	<b>Date</b> Date and time when the peak occurred.
String	<b>Value</b> The power consumption value (in watts).
Array	<b>History</b>
	<b>Object ConsumptionRecord</b>
DateTime	<b>Date</b> Date and time of the power consumption record.
String	<b>Value</b> The power consumption value (in watts).
	<b>Object ConsumptionRecord</b>
	.
	.
	<b>Object ConsumptionRecord</b>



## get-powerconsumptionsettings

Gets the current power consumption history configuration of a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] get-powerconsumptionsettings -ComputerName
<ComputerName> [-port <PortNumber>] [-Authentication <AuthenticationMode>]
-Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-LogLevel
<0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The `get-powerconsumptionsettings` command retrieves the current power consumption history configuration of a specific Lenovo ThinkServer System Manager. The configuration is comprised of the number of days the records are kept in the history and the interval in minutes between records.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### EXAMPLES

```
Invoke-TSMCLI -CommandName get-powerconsumptionsettings -ComputerName
myserver -port 80 -Authentication basic -Credential $cred
```

### RESPONSE

Object		Result
Integer	<b>Days</b>	Number of days to record.
Integer	<b>Interval</b>	The interval, in minutes, between power consumption records.

## get-powerstate

Gets the current power state of a specific ThinkServer.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] get-powerstate -ComputerName <ComputerName>
[-Port <PortNumber>] [-Authentication <AuthenticationMode>] -Credential
<credential> [-UseSSL [-SkipSSLCertificateCheck]] [-LogLevel <0|1|2|3|4|5|
6|7>]
```

### DESCRIPTION

The get-powerstate command gets the current power state on the ThinkServer specified by the provided Lenovo ThinkServer System Manager hostname/ip address and port. It returns the current power state and a numeric representation of it.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### EXAMPLES

```
Invoke-TSMCLI -CommandName get-powerstate -ComputerName myserver -Port 80
-Authentication basic -Credential $credential
```

### RESPONSE

Object NTPSettings		
Integer	<b>PowerState</b>	The numeric code for the server power state.
String	<b>PowerStateName</b>	The localized string for the server power state.

## get-psuredundancysettings

Gets the PSU redundancy settings of a specific ThinkServer.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] get-psuredundancysettings -ComputerName
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]
-Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-LogLevel
<0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The get-psuredundancysettings command gets the power supply redundancy settings of a specific ThinkServer.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### EXAMPLES

```
Invoke-TSMCLI -CommandName get-psuredundancysettings -ComputerName
myserver -Port 80 -Authentication basic -Credential $cred
```

### RESPONSE

Object PowerRedundancy							
Object	<p><b>UserRedundancyConfig</b></p> <table border="1"> <tr> <td>String</td> <td><b>Code</b></td> <td>The user defined power redundancy configuration code on the ThinkServer.</td> </tr> <tr> <td>String</td> <td><b>Description</b></td> <td>The user defined power redundancy configuration description on the ThinkServer.</td> </tr> </table>	String	<b>Code</b>	The user defined power redundancy configuration code on the ThinkServer.	String	<b>Description</b>	The user defined power redundancy configuration description on the ThinkServer.
String	<b>Code</b>	The user defined power redundancy configuration code on the ThinkServer.					
String	<b>Description</b>	The user defined power redundancy configuration description on the ThinkServer.					
Boolean	<p><b>UserPowerCappingEnabled</b>      The user defined power capping state.</p>						
Object	<p><b>CurrentRedundancyConfig</b></p> <table border="1"> <tr> <td>String</td> <td><b>Code</b></td> <td>The current power redundancy configuration code on the ThinkServer.</td> </tr> <tr> <td>String</td> <td><b>Description</b></td> <td>The current power redundancy configuration description on the ThinkServer.</td> </tr> </table>	String	<b>Code</b>	The current power redundancy configuration code on the ThinkServer.	String	<b>Description</b>	The current power redundancy configuration description on the ThinkServer.
String	<b>Code</b>	The current power redundancy configuration code on the ThinkServer.					
String	<b>Description</b>	The current power redundancy configuration description on the ThinkServer.					

Object <b>PSU 1 State</b>	String <b>Present</b>	Presence of PSU. "Present" if there is a PSU, "Absent" if no PSU is present.
	String <b>Input Power</b>	Presence of power on the input. "On" if power is detected. "Off" if not.
	String <b>Output Power</b>	State of the power output. "On" if PSU is turned on. "Off" if PSU is turned off.
Object <b>PSU 2 State</b>	String <b>Present</b>	Presence of PSU. "Present" if there is a PSU, "Absent" if no PSU is present.
	String <b>Input Power</b>	Presence of power on the input. "On" if power is detected. "Off" if not.
	String <b>Output Power</b>	State of the power output. "On" if PSU is turned on. "Off" if PSU is turned off.
Boolean <b>CurrentPowerCappingEnabled</b>	The current power capping state. "True" if power capping is enabled, "False" otherwise.	

## get-remotemediasettings

Gets the remote media settings and the list of available remote media images on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] get-remotemediasettings -ComputerName
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]
-Credential <credential> [-UseSSL [-SkipSSLCertificateCheck]] [-LogLevel
<0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The get-remotemediasettings command gets the remote media settings and the list of available remote media images on a specific Lenovo ThinkServer System Manager.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### EXAMPLES

```
Invoke-TSMCLI -CommandName get-remotemediasettings -ComputerName myserver
-Port 80 -Authentication basic -Credential $credential
```

### RESPONSE

Object RemoteMediaSettings								
Object	<b>Status</b>	<table border="1"> <tr> <td>Integer</td> <td><b>Code</b></td> <td>The numeric code for the Remote Media status.</td> </tr> <tr> <td>String</td> <td><b>Status</b></td> <td>The localized description of the Remote Media status.</td> </tr> </table>	Integer	<b>Code</b>	The numeric code for the Remote Media status.	String	<b>Status</b>	The localized description of the Remote Media status.
Integer	<b>Code</b>	The numeric code for the Remote Media status.						
String	<b>Status</b>	The localized description of the Remote Media status.						
String	<b>ServerAddress</b>	The address of the Remote Media server.						
String	<b>SourcePath</b>	The path to where the remote media is located on the server.						
String	<b>ShareType</b>	The type of sharing being used on the server (either "nfs" or "cifs").						
String	<b>Username</b>	The username to authenticate the share on the server (only for CIFS).						
String	<b>DomainName</b>	The domain name to authenticate the share on the server (only for CIFS).						

Array **RemoteMediaImages**

Object RemoteMediaImage											
String	<b>Type</b>	The type of remote media: "cddvd" for CD/DVD or "hd" for Hard Disk.									
String	<b>Name</b>	The name of the media file.									
Object	<b>Status</b>	<table border="1"> <tbody> <tr> <td>String</td> <td><b>RedirectionStatus</b></td> <td>The string that indicates the state of remote media redirection: "started", "inprogress" or "stopped".</td> </tr> <tr> <td>Integer</td> <td><b>StatusCode</b></td> <td>The numeric code indicating the status of the remote image redirection.</td> </tr> <tr> <td>String</td> <td><b>StatusDescription</b></td> <td>The localized description indicating the status of the remote image redirection.</td> </tr> </tbody> </table>	String	<b>RedirectionStatus</b>	The string that indicates the state of remote media redirection: "started", "inprogress" or "stopped".	Integer	<b>StatusCode</b>	The numeric code indicating the status of the remote image redirection.	String	<b>StatusDescription</b>	The localized description indicating the status of the remote image redirection.
String	<b>RedirectionStatus</b>	The string that indicates the state of remote media redirection: "started", "inprogress" or "stopped".									
Integer	<b>StatusCode</b>	The numeric code indicating the status of the remote image redirection.									
String	<b>StatusDescription</b>	The localized description indicating the status of the remote image redirection.									

Object RemoteMediaImage

...

Object RemoteMediaImage

## get-sel

Gets the SEL (System Event Log) of a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] get-sel -ComputerName <ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>] -Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-LogLevel <0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The get-sel command gets the SEL of the specified Lenovo ThinkServer System Manager.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### EXAMPLES

```
Invoke-TSMCLI -CommandName get-sel -ComputerName myserver -Port 80 -Authentication basic -Credential $cred
```

### RESPONSE

#### Array SEL

Object Event (Note that the fields described below may not be present on all SEL entries)		
Integer	<b>EventId</b>	The ID of the SEL event.
String	<b>EventType</b>	The type of the triggered event.
String	<b>Description</b>	The description of the event.
String	<b>SensorName</b>	The name of the sensor that triggered the event.
String	<b>Timestamp</b>	The timestamp of the event.
String	<b>SELData</b>	The RAW SEL data.

Object Event (Note that the fields described below may not be present on all SEL entries)

.  
.  
.

Object Event (Note that the fields described below may not be present on all SEL entries)

## get-sensors

Gets current sensor readings from a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] get-sensors -ComputerName <ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>] -Credential <credential> [-UseSSL [-SkipSSLCertificateCheck]] [[-CommandArgs] @ {sensorName="<sensor_name>"}] [-LogLevel <0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The get-sensors command gets the current sensor readings from a specific Lenovo ThinkServer System Manager. The list is ordered by name and contains name, type, unit, current reading and thresholds of each sensor. The type and unit codes are defined by the IPMI specification. If a sensor name is passed in, only information about that specific sensor is returned.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### sensorName

Optional argument that specifies the name of a sensor to get information from. If this argument is not specified, information about all sensors is returned.

### EXAMPLES

```
Invoke-TSMCLI -CommandName get-sensors -ComputerName myserver -Port 80 -Authentication basic -Credential $credential
```

```
Invoke-TSMCLI -CommandName get-sensors -ComputerName myserver -Port 80 -Authentication basic -Credential $credential -CommandArgs @ {sensorName="MySensorName" }
```

### RESPONSE

**Array Sensors (Not all sensors will have all the properties below)**

		Object Sensor
String	<b>Name</b>	The sensor unique name.



Object <b>Type</b>	<table border="1"> <tr> <td>Integer</td> <td><b>Code</b></td> <td>The numeric code for the sensor type.</td> </tr> <tr> <td>String</td> <td><b>Name</b></td> <td>The localized string for the sensor type.</td> </tr> </table>	Integer	<b>Code</b>	The numeric code for the sensor type.	String	<b>Name</b>	The localized string for the sensor type.
Integer	<b>Code</b>	The numeric code for the sensor type.					
String	<b>Name</b>	The localized string for the sensor type.					
Object <b>Unit</b>	<table border="1"> <tr> <td>Integer</td> <td><b>Code</b></td> <td>The numeric code for the sensor value unit.</td> </tr> <tr> <td>String</td> <td><b>Name</b></td> <td>The localized string for the sensor value unit.</td> </tr> </table>	Integer	<b>Code</b>	The numeric code for the sensor value unit.	String	<b>Name</b>	The localized string for the sensor value unit.
Integer	<b>Code</b>	The numeric code for the sensor value unit.					
String	<b>Name</b>	The localized string for the sensor value unit.					
Object <b>CurrentState</b>	<table border="1"> <tr> <td>String</td> <td><b>State</b></td> <td>The current state value of the sensor.</td> </tr> </table>	String	<b>State</b>	The current state value of the sensor.			
String	<b>State</b>	The current state value of the sensor.					
String <b>CurrentReading</b>	The current value reading of the sensor.						
String <b>LowerThresholdNonRecoverable</b>	The lower non-recoverable threshold for the sensor.						
String <b>LowerThresholdCritical</b>	The lower critical threshold for the sensor.						
String <b>LowerThresholdNonCritical</b>	The lower non-critical threshold for the sensor.						
String <b>UpperThresholdNonCritical</b>	The upper non-critical threshold for the sensor.						
String <b>UpperThresholdCritical</b>	The upper critical threshold for the sensor.						
String <b>UpperThresholdNonRecoverable</b>	The upper non-recoverable threshold for the sensor.						

**Object Sensor**

·  
·  
·

**Object Sensor**

## get-sensortypes

Gets the list of sensor types available on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] get-sensortypes -ComputerName <ComputerName>
[-Port <PortNumber>] [-Authentication <AuthenticationMode>] -Credential
<Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-LogLevel <0|1|2|3|4|
5|6|7>]
```

### DESCRIPTION

The get-sensortypes command gets the list of sensor types available on a specific Lenovo ThinkServer System Manager.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### EXAMPLES

```
Invoke-TSMCLI -CommandName get-sensortypes -ComputerName myserver -Port 80
-Authentication basic -Credential $cred
```

### RESPONSE

#### Array SensorTypes

Object SensorType		
Integer	<b>Code</b>	The numeric code of the sensor type.
String	<b>Description</b>	The localized description of the sensor type.

Object SensorType

·  
·  
·

Object SensorType

## get-services

Gets the settings for the services available on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] get-services -ComputerName <ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>] -Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-LogLevel <0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The get-services command retrieves the settings for the services available on a specific Lenovo ThinkServer System Manager.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### EXAMPLES

```
Invoke-TSMCLI -CommandName get-services -ComputerName myserver -Port 80 -Authentication basic -Credential $cred
```

### RESPONSE

#### Array Services

	Object Service
Integer	<b>ServiceId</b> The ID of the service.
String	<b>ServiceName</b> The name of the service.
Integer	<b>SecurePort</b> The number of the secure port. (Only if the service supports this configuration)
Integer	<b>NonsecurePort</b> The number of the non-secure port. (Only if the service supports this configuration)
Boolean	<b>Enabled</b> "True" if the specified service is enabled, "False" otherwise.

Object Service

·  
·  
·

Object Service

## get-smtpsettings

Gets the SMTP settings on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] get-smtpsettings -ComputerName <ComputerName>
[-Port <PortNumber>] [-Authentication <AuthenticationMode>] -Credential
<Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-LogLevel <0|1|2|3|4|
5|6|7>]
```

### DESCRIPTION

The get-smtpsettings command gets the SMTP settings on a specific Lenovo ThinkServer System Manager. This configuration is used by the TSM in order to send notifications by email.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### EXAMPLES

```
Invoke-TSMCLI -CommandName get-smtpsettings -ComputerName myserver -Port
80 -Authentication basic -Credential $cred
```

### RESPONSE

Object SMTPSettings																	
String	<b>SenderAddress</b>	The sender address for email messages.															
String	<b>MachineName</b>	The machine name for email messages.															
Object	<b>PrimarySMTPServer</b>	<table border="1"> <tbody> <tr> <td>Boolean</td> <td><b>SMTPSupportEnabled</b></td> <td>Indicates if the SMTP server is enabled.</td> </tr> <tr> <td>String</td> <td><b>ServerAddress</b></td> <td>The address of the SMTP server.</td> </tr> <tr> <td>Integer</td> <td><b>Port</b></td> <td>The port number of the SMTP server.</td> </tr> <tr> <td>Boolean</td> <td><b>RequiresAuthentication</b></td> <td>"True" if the server requires authentication, "False" otherwise.</td> </tr> <tr> <td>String</td> <td><b>UserName</b></td> <td>The username used in the SMTP server authentication.</td> </tr> </tbody> </table>	Boolean	<b>SMTPSupportEnabled</b>	Indicates if the SMTP server is enabled.	String	<b>ServerAddress</b>	The address of the SMTP server.	Integer	<b>Port</b>	The port number of the SMTP server.	Boolean	<b>RequiresAuthentication</b>	"True" if the server requires authentication, "False" otherwise.	String	<b>UserName</b>	The username used in the SMTP server authentication.
Boolean	<b>SMTPSupportEnabled</b>	Indicates if the SMTP server is enabled.															
String	<b>ServerAddress</b>	The address of the SMTP server.															
Integer	<b>Port</b>	The port number of the SMTP server.															
Boolean	<b>RequiresAuthentication</b>	"True" if the server requires authentication, "False" otherwise.															
String	<b>UserName</b>	The username used in the SMTP server authentication.															

Object **SecondarySMTPServer**

Boolean	<b>SMTPSupportEnabled</b>	Indicates if the SMTP server is enabled.
String	<b>ServerAddress</b>	The address of the SMTP server.
Integer	<b>Port</b>	The port number of the SMTP server.
Boolean	<b>RequiresAuthentication</b>	"True" if the server requires authentication, "False" otherwise.
String	<b>UserName</b>	The username used in the SMTP server authentication.

## get-solsettings

Gets the serial over LAN settings on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] get-solsettings -ComputerName <ComputerName>
[-Port <PortNumber>] [-Authentication <AuthenticationMode>] -Credential
<Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-LogLevel <0|1|2|3|4|
5|6|7>]
```

### DESCRIPTION

The get-solsettings command gets the serial over LAN (SOL) settings on a specific Lenovo ThinkServer System Manager. This configuration is comprised by the SOL enabled state, baud rate, flow control configuration and DTR hang up state.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### EXAMPLES

```
Invoke-TSMCLI -CommandName get-solsettings -ComputerName myserver -Port 80
-Authentication basic -Credential $cred
```

### RESPONSE

Object SOLSettings					
Boolean	<b>SOLEnabled</b> "TRUE" if SOL is enabled. "FALSE" otherwise.				
Integer	<b>BaudRate</b> The SOL baud rate.				
Object	<b>FlowControl</b> <table border="1" data-bbox="496 1543 1150 1632"> <tbody> <tr> <td>String</td> <td><b>Code</b> The SOL flow control code ("none", "xonxoff" or "hw").</td> </tr> <tr> <td>String</td> <td><b>Description</b> A localized string for the SOL flow control.</td> </tr> </tbody> </table>	String	<b>Code</b> The SOL flow control code ("none", "xonxoff" or "hw").	String	<b>Description</b> A localized string for the SOL flow control.
String	<b>Code</b> The SOL flow control code ("none", "xonxoff" or "hw").				
String	<b>Description</b> A localized string for the SOL flow control.				
Boolean	<b>DTRHangUp</b> "TRUE" if DTR hang up is enabled, "FALSE" otherwise.				

## get-sslcertificate

Gets the SSL certificate information on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] get-sslcertificate -ComputerName
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]
-Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-LogLevel
<0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The get-sslcertificate command gets the SSL certificate information on a specific Lenovo ThinkServer System Manager.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### EXAMPLES

```
Invoke-TSMCLI -CommandName get-sslcertificate -ComputerName myserver -Port
80 -Authentication basic -Credential $cred
```

### RESPONSE

Object CertificateSettings		
Object	<b>CertificateStatus</b>	
Boolean	<b>CertificateInstalled</b>	"True" if a certificate is installed, "False" otherwise.
DateTime	<b>CertificateInstallDateTime</b>	The date and time of the certificate installation.
DateTime	<b>PrivateKeyInstallDateTime</b>	The date and time of the private key installation.

<p>Object <b>CertificateInfo</b></p>	<p>Object <b>BasicInformation</b></p> <table border="1"> <tr> <td>String</td> <td><b>Version</b></td> <td>The version of the certificate.</td> </tr> <tr> <td>String</td> <td><b>SerialNumber</b></td> <td>The serial number of the certificate.</td> </tr> <tr> <td>String</td> <td><b>SignatureAlgorithm</b></td> <td>The signature algorithm used in the certificate.</td> </tr> <tr> <td>String</td> <td><b>PublicKey</b></td> <td>The public key of the certificate.</td> </tr> </table> <p>Object <b>IssuedFrom</b></p> <table border="1"> <tr> <td>String</td> <td><b>CommonName</b></td> <td>The common name of the certificate issuer.</td> </tr> <tr> <td>String</td> <td><b>Organization</b></td> <td>The organization of the certificate issuer.</td> </tr> <tr> <td>String</td> <td><b>OrganizationUnit</b></td> <td>The organizational unit of the certificate issuer.</td> </tr> <tr> <td>String</td> <td><b>CityOrLocality</b></td> <td>The city or location of the certificate issuer.</td> </tr> <tr> <td>String</td> <td><b>StateOrProvince</b></td> <td>The state or province of the certificate issuer.</td> </tr> <tr> <td>String</td> <td><b>Country</b></td> <td>The country of the certificate issuer.</td> </tr> <tr> <td>String</td> <td><b>Email</b></td> <td>The email of the certificate issuer.</td> </tr> </table> <p>Object <b>ValidityInformation</b></p> <table border="1"> <tr> <td>DateTime</td> <td><b>ValidFrom</b></td> <td>The certificate validity start date.</td> </tr> <tr> <td>DateTime</td> <td><b>ValidTo</b></td> <td>The certificate validity end date.</td> </tr> </table>	String	<b>Version</b>	The version of the certificate.	String	<b>SerialNumber</b>	The serial number of the certificate.	String	<b>SignatureAlgorithm</b>	The signature algorithm used in the certificate.	String	<b>PublicKey</b>	The public key of the certificate.	String	<b>CommonName</b>	The common name of the certificate issuer.	String	<b>Organization</b>	The organization of the certificate issuer.	String	<b>OrganizationUnit</b>	The organizational unit of the certificate issuer.	String	<b>CityOrLocality</b>	The city or location of the certificate issuer.	String	<b>StateOrProvince</b>	The state or province of the certificate issuer.	String	<b>Country</b>	The country of the certificate issuer.	String	<b>Email</b>	The email of the certificate issuer.	DateTime	<b>ValidFrom</b>	The certificate validity start date.	DateTime	<b>ValidTo</b>	The certificate validity end date.
String	<b>Version</b>	The version of the certificate.																																						
String	<b>SerialNumber</b>	The serial number of the certificate.																																						
String	<b>SignatureAlgorithm</b>	The signature algorithm used in the certificate.																																						
String	<b>PublicKey</b>	The public key of the certificate.																																						
String	<b>CommonName</b>	The common name of the certificate issuer.																																						
String	<b>Organization</b>	The organization of the certificate issuer.																																						
String	<b>OrganizationUnit</b>	The organizational unit of the certificate issuer.																																						
String	<b>CityOrLocality</b>	The city or location of the certificate issuer.																																						
String	<b>StateOrProvince</b>	The state or province of the certificate issuer.																																						
String	<b>Country</b>	The country of the certificate issuer.																																						
String	<b>Email</b>	The email of the certificate issuer.																																						
DateTime	<b>ValidFrom</b>	The certificate validity start date.																																						
DateTime	<b>ValidTo</b>	The certificate validity end date.																																						
<p>Object <b>CertificateInfo (continuation)</b></p>	<p>Object <b>IssuedTo</b></p> <table border="1"> <tr> <td>String</td> <td><b>CommonName</b></td> <td>The common name of the certificate holder.</td> </tr> <tr> <td>String</td> <td><b>Organization</b></td> <td>The organization of the certificate holder.</td> </tr> <tr> <td>String</td> <td><b>OrganizationUnit</b></td> <td>The organizational unit of the certificate holder.</td> </tr> <tr> <td>String</td> <td><b>CityOrLocality</b></td> <td>The city or location of the certificate holder.</td> </tr> <tr> <td>String</td> <td><b>StateOrProvince</b></td> <td>The state or province of the certificate holder.</td> </tr> <tr> <td>String</td> <td><b>Country</b></td> <td>The country of the certificate holder.</td> </tr> <tr> <td>String</td> <td><b>Email</b></td> <td>The email of the certificate holder.</td> </tr> </table>	String	<b>CommonName</b>	The common name of the certificate holder.	String	<b>Organization</b>	The organization of the certificate holder.	String	<b>OrganizationUnit</b>	The organizational unit of the certificate holder.	String	<b>CityOrLocality</b>	The city or location of the certificate holder.	String	<b>StateOrProvince</b>	The state or province of the certificate holder.	String	<b>Country</b>	The country of the certificate holder.	String	<b>Email</b>	The email of the certificate holder.																		
String	<b>CommonName</b>	The common name of the certificate holder.																																						
String	<b>Organization</b>	The organization of the certificate holder.																																						
String	<b>OrganizationUnit</b>	The organizational unit of the certificate holder.																																						
String	<b>CityOrLocality</b>	The city or location of the certificate holder.																																						
String	<b>StateOrProvince</b>	The state or province of the certificate holder.																																						
String	<b>Country</b>	The country of the certificate holder.																																						
String	<b>Email</b>	The email of the certificate holder.																																						



## get-systeminfo

Gets the system information of a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] get-systeminfo -ComputerName <ComputerName>
[-Port <PortNumber>] [-Authentication <AuthenticationMode>] -Credential
<Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-LogLevel <0|1|2|3|4|
5|6|7>]
```

### DESCRIPTION

The get-systeminfo command gets the system information of a specific Lenovo ThinkServer System Manager.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### EXAMPLES

```
Invoke-TSMCLI -CommandName get-systeminfo -ComputerName myserver -Port 80
-Authentication basic -Credential $cred
```

### RESPONSE

Object SystemInfo		
String	<b>ModelNumber</b>	The model number of the ThinkServer.
String	<b>ModelName</b>	The model name of the ThinkServer.
String	<b>TSMVersion</b>	The Lenovo ThinkServer System Manager current version.
String	<b>BIOSVersion</b>	The ThinkServer BIOS current version.
String	<b>SerialNumber</b>	The ThinkServer serial number.

## get-timezones

Gets all available timezones on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] get-timezones -ComputerName <ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>] -Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-LogLevel <0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The get-timezones command gets all available timezones on a specific Lenovo ThinkServer System Manager.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### EXAMPLES

```
Invoke-TSMCLI -CommandName get-timezones -ComputerName myserver -Port 80 -Authentication basic -Credential $cred
```

### RESPONSE

#### Array Timezones

Object Timezone		
String	<b>Timezone</b>	The id of the timezone.
String	<b>UTC</b>	The UTC offset of the timezone.

Object Timezone

·  
·  
·

Object Timezone

## get-users

Gets the list of local users on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] get-users -ComputerName <ComputerName> [-Port
<PortNumber>] [-Authentication <AuthenticationMode>] -Credential
<Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-LogLevel <0|1|2|3|4|
5|6|7>]
```

### DESCRIPTION

The get-users command gets the list of local users on a specific Lenovo ThinkServer System Manager. The list contains the id, username, e-mail address, role, KVM, virtual media, SNMP status, SNMP access, authentication protocol and privacy protocol for each user.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### EXAMPLES

```
Invoke-TSMCLI -CommandName get-users -ComputerName myserver -Port 80 -
Authentication basic -Credential $cred
```

### RESPONSE

#### Array Users

			Object User
Integer	<b>UserID</b>	The account user ID.	
String	<b>Username</b>	The account user name.	
String	<b>EmailAddress</b>	The email address for the user account.	
Object	<b>Role</b>		
		String <b>Code</b> The code for the user account role. Possible values are "admin", "user" or "operator".	
		String <b>Description</b> The localized description of the role.	
Boolean	<b>KVM</b>	"True" if the user account has KVM privilege, "False" otherwise.	
Boolean	<b>VirtualMedia</b>	"True" if the user account has Virtual Media privilege, "False" otherwise.	

Object	<b>SNMP</b>	
Boolean	<b>SNMPStatus</b>	"True" if the user account has SNMP enabled, "False" otherwise.
String	<b>SNMPAccess (Only available when SNMPStatus is "True")</b>	The SNMP access level. Either "read-only" or "read-write".
String	<b>AuthenticationProtocol (Only available when SNMPStatus is "True")</b>	The authentication protocol for SNMP settings. Either "sha" or "md5".
String	<b>PrivacyProtocol (Only available when SNMPStatus is "True")</b>	The encryption algorithm to use for SNMP settings. Either "des" or "aes".

Object User

.

.

.

Object User

## get-websessiontimeout

Gets the TSM Web UI session timeout (in seconds) that is configured on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] get-websessiontimeout -ComputerName  
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]  
-Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-LogLevel  
<0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The get-websessiontimeout command gets the TSM Web UI session timeout (in seconds) that is configured on a specific Lenovo ThinkServer System Manager.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### EXAMPLES

```
Invoke-TSMCLI -CommandName get-websessiontimeout -ComputerName myserver -  
Port 80 -Authentication basic -Credential $cred
```

### RESPONSE

<b>Integer WebSessionTimeout</b>

## reset-config

Restores a TSM to its factory configuration.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] reset-config -ComputerName <ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>] -Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-CommandArgs] @ {force="1"} [-LogLevel <0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The reset-config command restores a Lenovo ThinkServer System Manager to its factory configuration.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### force

The reset-config command will reset the TSM settings to their factory defaults, erasing any customizations already made. As a security measure, the "force" argument is required and must be specified as "1".

### EXAMPLES

```
Invoke-TSMCLI -CommandName reset-config -ComputerName myserver -Port 80 -Authentication basic -Credential $cred -CommandArgs @{force="1"}
```

### RESPONSE

Object Result		
Integer	<b>Result</b>	The response code for <b>Success</b> (always 0).
String	<b>Message</b>	The localized message for <b>Success</b> .

## reset-powerconsumptionhistory

Resets the power consumption history on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] reset-powerconsumptionhistory -ComputerName
<ComputerName> [-port <PortNumber>] [-Authentication <AuthenticationMode>]
-Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-
CommandArgs] @{force="1"} [-LogLevel <0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The reset-powerconsumptionhistory command deletes all power consumption history records on a specific Lenovo ThinkServer System Manager.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### force

The reset-powerconsumptionhistory command will delete all power consumption history records on a specific TSM. As a security measure, the "force" argument is required and must be specified as "1".

### EXAMPLES

```
Invoke-TSMCLI -CommandName reset-powerconsumptionhistory -ComputerName
myserver -port 80 -Authentication basic -Credential $cred @{force="1"}
```

### RESPONSE

Object Result		
Integer	<b>Result</b>	The response code for "Success" (always 0).
String	<b>Message</b>	The localized message for "Success".

## restore-config

Restores the configuration of a specific TSM from a local file.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] restore-config -ComputerName <ComputerName>
[-Port <PortNumber>] [-Authentication <AuthenticationMode>] -Credential
<Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-CommandArgs] @
{configFile="<path_to_image_file>"} [-LogLevel <0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The restore-config command restores the configuration of a specific Lenovo ThinkServer System Manager from a local file.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### configFile

The configuration file from a previous backup (required).

### EXAMPLES

```
Invoke-TSMCLI -CommandName restore-config -ComputerName myserver -Port 80
-Authentication basic -Credential $cred -CommandArgs @{configFile="C:
\myconfigfile"}
```

### RESPONSE

Object Result		
Integer	<b>Result</b>	The response code for Success (always 0).
String	<b>Message</b>	The localized message for Success.



## set-adrolegroup

Modifies an Active Directory role group on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] set-adrolegroup -ComputerName <ComputerName>
[-Port <PortNumber>] [-Authentication <AuthenticationMode>] -Credential
<credential> [-UseSSL [-SkipSSLCertificateCheck]] [-CommandArgs] @
{roleGroupId=<role_group_id>[;groupName="<group_name>"]
[;domainName="<domain_name>"] [;role="<admin|user|operator>"] [;kvm="<0|1>"]
[;virtualMedia="<0|1>"]} [-LogLevel <0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The set-adrolegroup command modifies an Active Directory role group on a specific Lenovo ThinkServer System Manager. Role groups map Active Directory groups to TSM roles, and are used to control which access level is granted to each group.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### roleGroupId

Specifies the role group identifier. To get the list of AD role groups, use the get-adrolegroups command.

#### groupName

The new name of the Active Directory group, with a maximum of 63 characters (optional). Hyphens and underscores are allowed.

#### domainName

The new name of the Active Directory domain, with a maximum of 255 characters (optional). A domain name must contain at least two labels (separated by a ".") and must not start with a ".". Each label can be up to 63 characters long. Hyphens are allowed, but not at the start or end of a label. Underscores are allowed, but not at the start of a label. Examples: "mycompany.com" and "mynetwork.localdomain".

#### role

Specifies the new group role (optional). Possible values are: "admin" for Administrator, "user" for User or "operator" for Operator.

#### kvm

Specifies the KVM privilege (optional).

#### virtualMedia

Specifies the Virtual Media privilege (optional).

## EXAMPLES

```
Invoke-TSMCLI -CommandName set-adrolegroup -ComputerName myserver -Port 80  
-Authentication basic -Credential $cred -CommandArgs @  
{roleGroupId=1;groupName="NewDomainOperators";domainName="newcompany.com"}
```

```
Invoke-TSMCLI -CommandName set-adrolegroup -ComputerName myserver -Port 80  
-Authentication basic -Credential $cred -CommandArgs @  
{roleGroupId=2;role="user";kvm="1"}
```

## RESPONSE

Object Result		
Integer	<b>Result</b>	The response code for "Success" (always 0).
String	<b>Message</b>	The localized message for "Success".

## set-adsettings

Sets the Active Directory settings on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] set-adsettings -ComputerName <ComputerName>
[-Port <PortNumber>] [-Authentication <AuthenticationMode>] -Credential
<credential> [-UseSSL [-SkipSSLCertificateCheck]] [-CommandArgs] @
{enabled=0} [-LogLevel <0|1|2|3|4|5|6|7>]
```

```
Invoke-TSMCLI [-CommandName] set-adsettings -ComputerName <ComputerName>
[-Port <PortNumber>] [-Authentication <AuthenticationMode>] -Credential
<credential> [-UseSSL [-SkipSSLCertificateCheck]] [-CommandArgs] @
{enabled=1
[;secretUser="<secret_user>";secretPassword="<secret_password>";domainName
e="<domain_name>";domainServer1="<ip_address>"[;domainServer2="<ip_address
>"[;domainServer3="<ip_address>"]]} [-LogLevel <0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The set-adsettings command sets the Active Directory settings on a specific Lenovo ThinkServer System Manager. This configuration is comprised of a status, a secret user and password, a domain name and up to three domain servers.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### enabled

Indicates if Active Directory should be enabled (1) or not (0).

#### secretUser

The user to be used to change the Active Directory settings (optional). It must contain 1 to 64 alpha-numeric characters, must start with an alphabetical character and is case sensitive. The following special characters are not allowed: ','(comma), '#'(pound sign), '.'(period), ':'(colon), ';'(semicolon), '\'(backslash), '['(square brackets), '<>(angle brackets), '|'(pipe), '='(equals), '+'(plus), '\*'(asterisk), '?'(question mark), ""(double quotes), ''(single quote) and '@'(at sign).

#### secretPassword

The password for the secret user (required if "secretUser" is specified). It must be at least 6 and at maximum 127 characters long. White spaces are not allowed.

#### domainName

The new Active Directory Domain (required if "enabled" is 1).

#### domainServer1

The first domain server (required if "enabled" is 1). It can be an IPv4 address or an IPv6 address.

**domainServer2**

The second domain server (optional). It can be an IPv4 address or an IPv6 address.

**domainServer3**

The third domain server (optional). It can be an IPv4 address or an IPv6 address.

**EXAMPLES**

```
Invoke-TSMCLI -CommandName set-adsettings -ComputerName myserver -Port 80
-Authentication basic -Credential $cred -CommandArgs @
{enabled=1;secretUser="user";secretPassword="123456";domainName="myDomainName.localdomain";domainServer1="192.168.1.104";domainServer2="192.168.8.111";domainServer3="fd6d:1983:eca4:ed90:0:0:0:1"}
```

```
Invoke-TSMCLI -CommandName set-adsettings -ComputerName myserver -Port 80
-Authentication basic -Credential $cred -CommandArgs @
{enabled=1;secretUser="user";secretPassword="123456";domainName="myDomainName.localdomain";domainServer1="192.168.1.104"}
```

```
Invoke-TSMCLI -CommandName set-adsettings -ComputerName myserver -Port 80
-Authentication basic -Credential $cred -CommandArgs @
{enabled=1;domainName="myDomainName.localdomain";domainServer1="192.168.1.104"}
```

```
Invoke-TSMCLI -CommandName set-adsettings -ComputerName myserver -Port 80
-Authentication basic -Credential $cred -CommandArgs @{enabled=0}
```

**RESPONSE**

Object Result	
Integer	<b>Result</b> The response code for "Success" (always 0).
String	<b>Message</b> The localized message for "Success".

## set-alertpolicy

Modifies an existing alert policy on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] set-alertpolicy -ComputerName <ComputerName>
[-Port <PortNumber>] [-Authentication <AuthenticationMode>] -Credential
<Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-CommandArgs] @
{alertPolicyId="<alert_policy_id>"[;policyNumber="<policy_number>"]
[;policyEnabled="<0|1>"] [;policySet="<policy_set>"]
[;lanDestinationId="<lan_destination_id>"] [;eventSpecificString="<0|1>"]
[;alertStringKey="<alert_string_key>"]} [-LogLevel <0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The set-alertpolicy command modifies an Alert Policy on a specific Lenovo ThinkServer System Manager. At least one optional command argument must be provided.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### alertPolicyId

The id of the Alert Policy to be modified (required). To get the list of valid alertPolicyIds on the TSM, use the command "get-alertpolicies".

#### policyNumber

The policy number that will be used in the event filters. Valid values are 1 to 15.

#### policyEnabled

Indicates if the Alert Policy should be enabled (1) or not (0).

#### policySet

The policy set number. It can be:

0 - Always send alert to this destination;

1 - If alert to previous destination was successful, do not send alert to this destination. Proceed to next entry in this policy set;

2 - If alert to previous destination was successful, do not send alert to this destination. Do not process any more entries in this policy set;

3 - If alert to previous destination was successful, do not send alert to this destination. Proceed to next entry in this policy set that is to a different channel;

4 - If alert to previous destination was successful, do not send alert to this destination. Proceed to next entry in this policy set that is to a different destination type.

**lanDestinationId**

The LAN Destination of the policy. To get the list of available LAN destinations, invoke the TSMCLI command "get-landestinations".

**eventSpecificString**

Indicates if an event specific alert string should be used (1) or not (0).

**alertStringKey**

The alert string key contained in the PEF configuration parameters, to specify which string is to be sent for this Alert Policy (required if eventSpecificString is changed from "0" to "1"). Valid values are between the range 0 to 127.

**EXAMPLES**

```
Invoke-TSMCLI -CommandName set-alertpolicy -ComputerName myserver -Port 80
-Authentication basic -Credential $cred -CommandArgs @
{alertPolicyId="1";policyNumber="3";policyEnabled="1";policySet="2";lanDes
tinationId="4";eventSpecificString="1";alertStringKey="15"}
```

**RESPONSE**

Object Result		
Integer	<b>Result</b>	The response code for "Success" (always 0).
String	<b>Message</b>	The localized message for "Success".

## set-auditlogsettings

Enables or disables the audit log on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] set-auditlogsettings -ComputerName
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]
-Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-
CommandArgs] @{auditLogEnabled="<0|1>"} [-LogLevel <0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The set-auditlogsettings command enables or disables the audit log on a specific Lenovo ThinkServer System Manager.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### auditLogEnabled

Enables the audit log when the argument value is 1 and disables when 0 (required).

### EXAMPLES

```
Invoke-TSMCLI -CommandName set-auditlogsettings -ComputerName myserver -
Port 80 -Authentication basic -Credential $cred -CommandArgs @
{auditLogEnabled="1"}
```

### RESPONSE

Object Result		
Integer	<b>Result</b>	The response code for "Success" (always 0).
String	<b>Message</b>	The localized message for "Success".

## set-datetime

Sets the date and time on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] set-datetime -ComputerName <ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>] -Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-CommandArgs] @ {dateTime="<YYYY-MM-DDThh:mm[:ss]>"|<DateTimeObject>} [-LogLevel <0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The set-datetime command sets the date and time on a specific Lenovo ThinkServer System Manager. This configuration is comprised of an ISO 8601 formatted date and time (without the timezone information). Note that running this command will disable the automatic synchronization with an NTP server.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### dateTime

Specifies the new date and time. It must be either a formatted date and time in the format "YYYY-MM-DDThh:mm[:ss]" (seconds are optional) or a DateTime PowerShell object. Valid date and time range is from 2005-01-01T00:00:00 UTC to 2038-01-19T03:14:07 UTC.

### EXAMPLES

```
Invoke-TSMCLI -CommandName set-datetime -ComputerName myserver -Port 80 -Authentication basic -Credential $cred -CommandArgs @ {dateTime="2013-11-16T19:20:30"}
```

```
Invoke-TSMCLI -CommandName set-datetime -ComputerName myserver -Port 80 -Authentication basic -Credential $cred -CommandArgs @ {dateTime="2013-11-16T19:20"}
```

```
Invoke-TSMCLI -CommandName set-datetime -ComputerName myserver -Port 80 -Authentication basic -Credential $cred -CommandArgs @{dateTime=(Get-Date)}
```

```
Invoke-TSMCLI -CommandName set-datetime -ComputerName myserver -Port 80 -Authentication basic -Credential $cred -CommandArgs @{dateTime=(Get-Date "2014-08-14 09:16:25")}
```



```
Invoke-TSMCLI -CommandName set-datetime -ComputerName myserver -Port 80 -  
Authentication basic -Credential $cred -CommandArgs @{dateTime=New-Object  
Datetime(2014, 3, 10, 8, 30, 14)}
```

## RESPONSE

Object Result	
Integer	<b>Result</b> The response code for "Success" (always 0).
String	<b>Message</b> The localized message for "Success".

## set-dns

Configures DNS settings on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] set-dns -ComputerName <ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>] -Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-CommandArgs] @ {mode="manual";dnsServer1="<server_ip_address>"[;dnsServer2="<server_ip_address>"[;dnsServer3="<server_ip_address>"]} [-LogLevel <0|1|2|3|4|5|6|7>]
```

```
Invoke-TSMCLI [-CommandName] set-dns -ComputerName <ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>] -Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-CommandArgs] @ {mode="auto";ipPriority="<ipv4|ipv6>"} [-LogLevel <0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The set-dns command configures which DNS servers are used by a specific Lenovo ThinkServer System Manager.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### mode

Specifies if the DNS server addresses should be set automatically using DHCP, or if they should be manually configured. Possible values are "auto" and "manual".

#### dnsServer1

IPv4 or IPv6 address of the first DNS server. Mandatory if mode is "manual".

#### dnsServer2

IPv4 or IPv6 address of the second DNS server (optional).

#### dnsServer3

IPv4 or IPv6 address of the third DNS server (optional).

#### ipPriority

If set to "ipv4", there will be 2 IPv4 DNS servers and 1 IPv6 DNS server. If set to "ipv6", there will be 2 IPv6 DNS servers and 1 IPv4 DNS server. Mandatory if mode is "auto".

## EXAMPLES

```
Invoke-TSMCLI -CommandName set-dns -ComputerName myserver -Port 80 -
Authentication basic -Credential $cred -CommandArgs @
{mode="manual";dnsServer1="192.168.1.104";dnsServer2="192.168.1.254";dnsSe
rver3="192.168.10.254"}
```

```
Invoke-TSMCLI -CommandName set-dns -ComputerName myserver -Port 80 -
Authentication basic -Credential $cred -CommandArgs @
{mode="manual";dnsServer1="192.168.1.104"}
```

```
Invoke-TSMCLI -CommandName set-dns -ComputerName myserver -Port 80 -
Authentication basic -Credential $cred -CommandArgs @
{mode="auto";ipPriority="ipv4"}
```

## RESPONSE

Object Result		
Integer	<b>Result</b>	The response code for "Success" (always 0).
String	<b>Message</b>	The localized message for "Success".

## set-eventfilter

Modifies an existing Event Filter on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] set-eventfilter -ComputerName <ComputerName>
[-Port <PortNumber>] [-Authentication <AuthenticationMode>] -Credential
<Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-CommandArgs] @
{eventFilterId="<event_filter_id>";[eventFilterEnabled="<0|1>";]
[severity="<unspecified|monitor|information|normal|non-critical|critical|
non-recoverable>";][powerAction="<none|power-down|power-reset|power-
cycle>";][policyNumber="<1-15>";][<GENERATOR ID ARGUMENTS>;][<SENSOR
ARGUMENTS>;][<EVENT DATA ARGUMENTS>}] [-LogLevel <0|1|2|3|4|5|6|7>]
```

#### GENERATOR ID ARGUMENTS

There are 3 different ways of specifying the generatorId arguments:

1. By setting generatorIdRawData="1" and also specifying generatorId1="<hex\_value>";generatorId2="<hex\_value>"
2. By setting generatorIdRawData="0" and eventGenerator="slave-address" and also specifying generator="<slave\_address>";generatorChannelNumber="<generator\_channel\_number>";generatorIPMBDeviceLUN="<generator\_ipmb\_device\_lun>"
3. By setting generatorIdRawData="0" and eventGenerator="software-id" and also specifying generator="<software\_id>";generatorChannelNumber="<generator\_channel\_number>"

All three above methods produce values for generatorId1 and generatorId2, which will be displayed by the TSMCLI command "get-eventfilters" when visualizing the information of the corresponding event filter.

#### SENSOR ARGUMENTS

```
sensorMode="all"
```

```
sensorMode="sensor-type";sensorType="<sensor_type>";eventOptions="all"
```

```
sensorMode="sensor-type";sensorType="<sensor_type>";eventOptions="sensor-
events";sensorEvents="<sensor-event-1,sensor-event-2,...,sensor-event-n>"
```

```
sensorMode="sensor-name";sensorName="<sensor_name>";eventOptions="all"
```

```
sensorMode="sensor-name";sensorName="<sensor_name>";eventOptions="sensor-
events";sensorEvents="<sensor-event-1,sensor-event-2,...,sensor-event-n>"
```

#### EVENT DATA ARGUMENTS

```
eventTrigger="<event_trigger>"[;eventData1ANDMask="<event_data_1_and_mask>"]
[;eventData1Compare1="<event_data_1_compare_1>"]
[;eventData1Compare2="<event_data_1_compare_2>"]
[;eventData2ANDMask="<event_data_2_and_mask>"]
[;eventData2Compare1="<event_data_2_compare_1>"]
[;eventData2Compare2="<event_data_2_compare_2>"]
[;eventData3ANDMask="<event_data_3_and_mask>"]
[;eventData3Compare1="<event_data_3_compare_1>"]
[;eventData3Compare2="<event_data_3_compare_2>"]
```

## DESCRIPTION

The `set-eventfilter` command modifies an Event Filter on a specific Lenovo ThinkServer System Manager.

## PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

## COMMAND ARGUMENTS

### **eventFilterId**

The id of the Event Filter to be modified (required).

### **eventFilterEnabled**

Indicates if the Event Filter should be enabled (1) or not (0).

### **severity**

The minimum event severity to filter. The Event Filter severity can be "unspecified", "monitor", "information", "normal", "non-critical", "critical" or "non-recoverable".

### **powerAction**

Selects the power action to be executed. It can be "none", "power-down", "power-reset" or "power-cycle".

### **policyNumber**

Specifies the policy number related to this event filter (required). It allows to trigger alert policies associated with this policy number. Valid values are 1 to 15.

### **generatorIdRawData**

Indicates if the Event Filter should use a generator from raw data (1) or from specified generator (0).

### **eventGenerator**

Indicates if the event generator should be an I2C slave address or a software ID (required if `generatorIdRawData` is "0"). It can be "slave-address" or "software-id".

### **generator**

The generator I2C address or software ID (required if `generatorIdRawData` is "0"). Valid values are 0 to 127.

**generatorChannelNumber**

The particular channel number through which the event message is received over (required if generatorIdRawData is "0"). Choose '0' if the event message is received via the system interface, primary IPMB, or internally generated by the TSM. Valid values are 0 to 15.

**generatorIPMBDeviceLUN**

The corresponding IPMB device LUN if event generated by IPMB (required if eventGenerator is "slave-address"). Valid values are 1 to 3.

**generatorID1**

The raw generator ID1 data value (required if generatorIdRawData is "1"). Valid values are 0x0 to 0xFF.

**generatorID2**

The raw generator ID2 data value (required if generatorIdRawData is "1"). Valid values are 0x0 to 0xFF.

**sensorMode**

The sensor selection mode, it can be:

all - to all sensors;

sensor-type - to select the sensors by it's type ID;

sensor-name - to select the sensor by name.

**sensorType**

The sensor type identification (required if sensorMode is "sensor-type"). To get the list of available sensor types, invoke the TSMCLI command "get-sensortypes".

**sensorName**

The sensor name (required if sensorMode is "sensor-name"). To get the list of available sensor names, invoke the TSMCLI command "get-sensors".

**eventOptions**

Specifies the events from the selected sensors (required if sensorMode is different from "all"). Possible values are: "all" for all events, "sensor-events" for sensor specific events.

**sensorEvents**

A comma separated list of all possible events for the selected sensors (required if eventOptions is "sensor-events"). Each event can be:

lower-non-critical-going-low;

lower-non-critical-going-high;

lower-critical-going-low;

lower-critical-going-high;

lower-non-recoverable-going-low;

lower-non-recoverable-going-high;

upper-non-critical-going-low;  
upper-non-critical-going-high;  
upper-critical-going-low;  
upper-critical-going-high;  
upper-non-recoverable-going-low;  
upper-non-recoverable-going-high.

**eventTrigger**

This event/reading type value. Valid values are 1 to 255.

**eventData1ANDMask**

Indicates wildcarded or compared bits. Valid values are 0 to 255.

**eventData1Compare1**

Indicates whether each bit position's comparison is an exact comparison or not. Valid values are 0 to 255.

**eventData1Compare2**

Indicates whether each bit position's comparison is an exact comparison or not. Valid values are 0 to 255.

**eventData2ANDMask**

Indicates wildcarded or compared bits. Valid values are 0 to 255.

**eventData2Compare1**

Indicates whether each bit position's comparison is an exact comparison or not. Valid values are 0 to 255.

**eventData2Compare2**

Indicates whether each bit position's comparison is an exact comparison or not. Valid values are 0 to 255.

**eventData3ANDMask**

Indicates wildcarded or compared bits. Valid values are 0 to 255.

**eventData3Compare1**

Indicates whether each bit position's comparison is an exact comparison or not. Valid values are 0 to 255.

**eventData3Compare2**

Indicates whether each bit position's comparison is an exact comparison or not. Valid values are 0 to 255.

**EXAMPLES**

```
Invoke-TSMCLI -CommandName set-eventfilter -ComputerName myserver -Port 80  
-Authentication basic -Credential $cred -CommandArgs @
```

```
{eventFilterId=2;eventFilterEnabled=1;severity="non-critical";powerAction="none";policyNumber=4;generatorIdRawData="1";generatorId1="0xFF";generatorId2="0xFF";sensorMode="sensor-name";sensorName="Fan_4-1";eventOptions="all";eventTrigger=255;eventData1ANDMask=0;eventData1Compare1=0;eventData1Compare2=0;eventData2ANDMask=2;eventData2Compare1=2;eventData2Compare2=0;eventData3ANDMask=1;eventData3Compare1=1;eventData3Compare2=1}
```

```
Invoke-TSMCLI -CommandName set-eventfilter -ComputerName myserver -Port 80 -Authentication basic -Credential $cred -CommandArgs @
{eventFilterId=3;generatorIdRawData="1";generatorId1="0xFF";generatorId2="0xFF";sensorMode="sensor-name";sensorName="Fan_4-1";eventOptions="sensor-events";sensorEvents="lower-non-critical-going-low,lower-non-critical-going-high,upper-non-critical-going-low"}
```

```
Invoke-TSMCLI -CommandName set-eventfilter -ComputerName myserver -Port 80 -Authentication basic -Credential $cred -CommandArgs @
{eventFilterId=2;eventData3ANDMask=1;eventData3Compare1=1;eventData3Compare2=1}
```

## RESPONSE

Object Result		
Integer	<b>Result</b>	The response code for <b>Success</b> (always 0).
String	<b>Message</b>	The localized message for <b>Success</b> .



## set-identifierstate

Set the state of physical system identification.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] set-identifierstate -ComputerName
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]
-Credential <credential> [-UseSSL [-SkipSSLCertificateCheck] [-
CommandArgs] @{ operation=<"on"|"off"> } [-LogLevel <0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The set-identifierstate command set the state of physical system identification, typically a LED.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### operation

Specifies the state to be set on the physical system identification of the ThinkServer. Possible values are: "on" to turn on the ThinkServer identification and "off" to immediately turn off.

### EXAMPLES

```
Invoke-TSMCLI -CommandName set-identifierstate -ComputerName myserver -Port 80 -
Authentication basic -Credential $credential -CommandArgs @{ operation="on" }
```

### RESPONSE

Object Result		
Integer	<b>Result</b>	The response code for "Success" (always 0).
String	<b>Message</b>	The localized message for "Success".

## set-landestination

Modifies a LAN Destination on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] set-landestination -ComputerName
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]
-Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-
CommandArgs] @
{lanDestinationId="<lan_destination_id>";type="snmp";destinationAddress="<
ip_address>"} [-LogLevel <0|1|2|3|4|5|6|7>]
```

```
Invoke-TSMCLI [-CommandName] set-landestination -ComputerName
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]
-Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-
CommandArgs] @
{lanDestinationId="<lan_destination_id>";type="email";userId="<user_id>"}
[-LogLevel <0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The set-landestination command modifies a LAN Destination on a specific Lenovo ThinkServer System Manager.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### lanDestinationId

The id of the LAN Destination to be modified (required).

#### type

The type of the LAN Destination to be modified (required). The possible values are "snmp" if the destination is snmp trap or "email" if is an e-mail.

#### destinationAddress

The IPv4 or IPv6 address to where SNMP notifications will be sent (required if type is "snmp"). Provided IPv4 addresses must not start with 0.

#### userId

The user to whom email alerts will be sent (required if type is "email"). Note that an email address must be configured for the selected user account. To get the registered users, invoke the TSMCLI command "get-users".

## EXAMPLES

```
Invoke-TSMCLI -CommandName set-landestination -ComputerName myserver -Port
80 -Authentication basic -Credential $cred -CommandArgs @
{lanDestinationId="6";type="snmp";destinationAddress="192.168.1.10"}
```

```
Invoke-TSMCLI -CommandName set-landestination -ComputerName myserver -Port
80 -Authentication basic -Credential $cred -CommandArgs @
{lanDestinationId="6";type="email";userId="2"}
```

## RESPONSE

Object Result		
Integer	<b>Result</b>	The response code for "Success" (always 0).
String	<b>Message</b>	The localized message for "Success".

## set-ldaprolegroup

Modifies an LDAP role group on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] set-ldaprolegroup -ComputerName
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]
-Credential <credential> [-UseSSL [-SkipSSLCertificateCheck]] [-
CommandArgs] @{roleGroupId=<role_group_id>[;groupName="<group_name>"]
[;groupSearchBase="<group_search_base>"] [;role="<admin|user|operator>"]
[;kvm="<0|1>"] [;virtualMedia="<0|1>"]} [-LogLevel <0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The set-ldaprolegroup command modifies an LDAP role group on a specific Lenovo ThinkServer System Manager. Role groups map LDAP groups to TSM roles, and are used to control which access level is granted to each group.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### roleGroupId

Specifies the role group identifier. To get the list of LDAP role groups, use the get-ldaprolegroups command.

#### groupName

The new name of the LDAP group, with a maximum of 63 characters (optional). Hyphens and underscores are allowed.

#### groupSearchBase

The new search base path from where the role group is located on the LDAP server (optional). It shall be at least 4 and at most 63 alpha-numeric characters long. Hyphens, dots and underscores are allowed. Examples are "dc=corp,dc=domain,dc=com" and "dc=people,dc=mydept,dc=mycompany,dc=com".

#### role

Specifies the new group role (optional). Possible values are: "admin" for Administrator, "user" for User or "operator" for Operator.

#### kvm

Specifies the KVM privilege (optional).

#### virtualMedia

Specifies the Virtual Media privilege (optional).

## EXAMPLES

```
Invoke-TSMCLI -CommandName set-ldaprolegroup -ComputerName myserver -Port
80 -Authentication basic -Credential $cred -CommandArgs @
{roleGroupId=1;groupName="NewDomainOperators";groupSearchBase="dc=corp,dc=
domain,dc=com"}
```

```
Invoke-TSMCLI -CommandName set-ldaprolegroup -ComputerName myserver -Port
80 -Authentication basic -Credential $cred -CommandArgs @
{roleGroupId="2";role="user";kvm="1"}
```

## RESPONSE

Object Result		
Integer	<b>Result</b>	The response code for "Success" (always 0).
String	<b>Message</b>	The localized message for "Success".

## set-ldapsettings

Sets the LDAP settings on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] set-ldapsettings -ComputerName <ComputerName>
[-Port <PortNumber>] [-Authentication <AuthenticationMode>] -Credential
<credential> [-UseSSL [-SkipSSLCertificateCheck]] [-CommandArgs] @
{enabled=0} [-LogLevel <0|1|2|3|4|5|6|7>]
```

```
Invoke-TSMCLI [-CommandName] set-ldapsettings -ComputerName <ComputerName>
[-Port <PortNumber>] [-Authentication <AuthenticationMode>] -Credential
<credential> [-UseSSL [-SkipSSLCertificateCheck]] [-CommandArgs] @
{enabled=1;encryptedType="<noencrypted|
ssl>";serverAddress="<server_address>";port="<port_number>";bindDN="<bind_
dn>";password="<password>";searchBase="<search_base>";attributeOfUserLogin
="<cn|uid>"} [-LogLevel <0|1|2|3|4|5|6|7>]
```

```
Invoke-TSMCLI [-CommandName] set-ldapsettings -ComputerName <ComputerName>
[-Port <PortNumber>] [-Authentication <AuthenticationMode>] -Credential
<credential> [-UseSSL [-SkipSSLCertificateCheck]] [-CommandArgs] @
{enabled=1;encryptedType="<starttls>";commonNameType="<ipaddress|
fqdn>";serverAddress="<server_address>";port="<port_number>";bindDN="<bind_
dn>";password="<password>";searchBase="<search_base>";attributeOfUserLogi
n="<cn|uid>"[;caCertFile="<path_to_certificate_file>"]
[;certFile="<path_to_certificate_file>"]
[;privateKeyFile="<path_to_private_key_file>"]} [-LogLevel <0|1|2|3|4|5|6|
7>]
```

### DESCRIPTION

The set-ldapsettings command sets the LDAP settings on a specific Lenovo ThinkServer System Manager.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### enabled

Indicates if LDAP should be enabled (1) or not (0).

#### encryptedType

Encryption type. The possible values are "noencrypted", "ssl" and "starttls".

#### commonNameType

The common name type. Required if encryptedType is set to "starttls". The possible values are "ipaddress" and "fqdn".

### **serverAddress**

The LDAP server address. If common name type is set to "fqdn", serverAddress can be an IPv4 address or a fully qualified domain name. Otherwise, it can be an IPv4 address or an IPv6 address.

### **port**

The port used to start an LDAP session. It shall be a value from 1 to 65535.

### **bindDN**

The LDAP Bind DN with a maximum of 63 alpha-numeric characters. There shall be at least 2 DNs specified and the first DN must be "cn" or "uid". Examples are "cn=admin,dc=corp,dc=domain,dc=com" and "cn=admin,dc=people,dc=mydept,dc=mycompany,dc=com".

### **password**

LDAP Bind password. It shall be at least 1 and at maximum 47 characters long. ' '(whitespace) is not allowed.

### **searchBase**

LDAP search base. It shall be at least 4 and at most 63 alpha-numeric characters long. Hyphens, dots and underscores are allowed. Examples are "dc=corp,dc=domain,dc=com" and "dc=people,dc=mydept,dc=mycompany,dc=com".

### **attributeOfUserLogin**

Tells the LDAP server which attribute should be used to identify the user. Supported values are "cn" or "uid".

### **caCertFile**

CA certificate file to upload to the TSM (optional). This argument can be provided when the encryption type is set to "StartTLS".

### **certFile**

The certificate file to upload to the TSM (optional). This argument can be provided when the encryption type is set to "StartTLS".

### **privateKeyFile**

The private key file to upload to the TSM (optional). This argument can be provided when the encryption type is set to "StartTLS".

## **EXAMPLES**

```
Invoke-TSMCLI -CommandName set-ldapsettings -ComputerName myserver -Port
80 -Authentication basic -Credential $cred -CommandArgs @
{enabled=1;encryptedType="noencrypted";serverAddress="192.168.1.102";port=
369;bindDN="cn=admin,dc=corp,dc=domain,dc=com";password="1234";searchBase=
"dc=corp,dc=domain,dc=com";attributeOfUserLogin="cn"}
```

```
Invoke-TSMCLI -CommandName set-ldapsettings -ComputerName myserver -Port
80 -Authentication basic -Credential $cred -CommandArgs @
```

```
{enabled=1;encryptedType="starttls";commonNameType="ipaddress";serverAddress="192.168.1.102";port="389";bindDN="cn=admin,dc=corp,dc=domain,dc=com";password="1234";searchBase="dc=corp,dc=domain,dc=com";attributeOfUserLogin="cn";caCertFile="C:\MyCertFolder\tsm_cacert.pem";certFile="C:\MyCertFolder\tsm_cert.pem";privateKeyFile="C:\MyCertFolder\tsm_key.pem"}
```

```
Invoke-TSMCLI -CommandName set-ldapsettings -ComputerName myserver -Port 80 -Authentication basic -Credential $cred -CommandArgs @{enabled=0}
```

## RESPONSE

Object Result		
Integer	<b>Result</b>	The response code for "Success" (always 0).
String	<b>Message</b>	The localized message for "Success".



## set-niclinksettings

Sets the network link settings on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] set-niclinksettings -ComputerName
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]
-Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-
CommandArgs] @{mode="auto"} [-LogLevel <0|1|2|3|4|5|6|7>]
```

```
Invoke-TSMCLI [-CommandName] set-niclinksettings -ComputerName
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]
-Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-
CommandArgs] @{mode="manual";fullDuplex="<1|0>";linkSpeed="<10|100>"} [-
LogLevel <0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The set-niclinksettings command sets the network link configuration on a specific Lenovo ThinkServer System Manager. This configuration is formed by the link mode, link speed and duplex mode settings and can only be changed if the network interface mode is dedicated.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### mode

Specifies the negotiation mode for the link settings of the device (required). The possible values are "auto" to the device perform automatic configuration or "manual" if the new settings will be defined manually.

#### fullDuplex

Enables the Full Duplex when the argument is 1 and disables when 0 (required if changing the negotiation mode to "manual").

#### linkSpeed

Specifies the link speed in Mbps (required if changing the negotiation mode to "manual"). Valid values are 10 and 100.

### EXAMPLES

```
Invoke-TSMCLI -CommandName set-niclinksettings -ComputerName myserver -
Port 80 -Authentication basic -Credential $cred -CommandArgs @
{mode="auto"}
```

```
Invoke-TSMCLI -CommandName set-niclinksettings -ComputerName myserver -  
Port 80 -Authentication basic -Credential $cred -CommandArgs @  
{mode="manual";fullDuplex="1";linkSpeed="100"}
```

## RESPONSE

Object		Result
Integer	<b>Result</b>	The response code for Success (always 0).
String	<b>Message</b>	The localized message for Success.

## set-nicsettings

Sets the network configuration of a specific network interface on the TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] set-nicsettings -ComputerName <ComputerName>
[-Port <PortNumber>] [-Authentication <AuthenticationMode>] -Credential
<credential> [-UseSSL [-SkipSSLCertificateCheck]] [-CommandArgs] @
{[nicMode="<shared|dedicated>"] [;<IPV4 SETTINGS>] [;<IPV6 SETTINGS>] [;<VLAN
SETTINGS>]} [-LogLevel <0|1|2|3|4|5|6|7>]
```

IPV4 SETTINGS (required if changing the network mode)

```
ipv4Mode="dhcp"
```

```
ipv4Mode="static";ipv4Address="<ip_address>";ipv4SubnetMask="<subnet_mask>";
ipv4DefaultGateway="<default_gateway>"
```

IPV6 SETTINGS (disabled by default when changing the network mode)

```
ipv6Enabled="0"
```

```
ipv6Enabled="1";ipv6Mode="dhcp"
```

```
ipv6Enabled="1";ipv6Mode="static";ipv6Address="<ip_address>";ipv6SubnetPre
fixLength="<subnet_prefix_length>";ipv6DefaultGateway="<default_gateway>"
```

VLAN SETTINGS (disabled by default when changing the network mode)

```
vlanEnabled="0"
```

```
vlanEnabled="1";vlanId="<vlan_id>";vlanPriority="<vlan_priority>"
```

### DESCRIPTION

The set-nicsettings command sets the network configuration on the Lenovo ThinkServer System Manager. When switching between the NIC modes (shared/dedicated) this operation can take some minutes. During this time, network connectivity to the TSM can be lost.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### nicMode

Specifies the mode for network interface. The available modes are "shared" or "dedicated" (optional). When changing the NIC mode the IPv4 arguments are required.

#### ipv4Mode

Specifies the IPv4 assignment mode. The possible values are "dhcp" if the configuration will be retrieved from a DHCP server or "static" if the new settings will be defined manually.

**ipv4Address (required if IPv4 mode is changed from "dhcp" to "static")**

Specifies the new IPv4 address for the network interface.

**ipv4SubnetMask (required if IPv4 mode is changed from "dhcp" to "static")**

Specifies the new network subnet mask for the network interface.

**ipv4DefaultGateway (required if IPv4 mode is changed from "dhcp" to "static")**

Specifies the new default IPv4 gateway for the network interface.

**ipv6Enabled**

Specifies whether the IPv6 settings should be enabled (1) or not (0). When changing the NIC mode the IPv6 settings will be disabled by default.

**ipv6Mode**

Specifies the IPv6 assignment mode. The possible values are "dhcp" if the configuration will be retrieved from a DHCP server or "static" if the new settings will be defined manually.

**ipv6Address (required if IPv6 mode is changed from "dhcp" to "static")**

Specifies the new IPv6 address for the network interface.

**ipv6SubnetPrefixLength (required if IPv6 mode is changed from "dhcp" to "static")**

Specifies the new network subnet prefix length for the network interface. This value must be a number between 0 and 128.

**ipv6DefaultGateway (required if IPv6 mode is changed from "dhcp" to "static")**

Specifies the new default IPv6 gateway for the network interface.

**vlanEnabled**

Specifies whether the VLAN should be enabled (1) or not (0) for this NIC. When changing the NIC mode the VLAN will be disabled by default.

**vlanId (required if vlanEnabled is "1")**

Specifies the id of the VLAN (2 to 4094).

**vlanPriority (required if vlanEnabled is "1")**

Specifies the priority of the VLAN (0 to 7).

## EXAMPLES

```
Invoke-TSMCLI -CommandName set-nicsettings -ComputerName myserver -Port 80
-Authentication basic -Credential $credential -CommandArgs @
{ipv4Mode="static";ipv4Address="192.168.1.12";ipv4SubnetMask="255.255.255.
0";ipv4DefaultGateway="192.168.1.1"}
```

```
Invoke-TSMCLI -CommandName set-nicsettings -ComputerName myserver -Port 80
-Authentication basic -Credential $credential -CommandArgs @
{ipv6Mode="static";ipv6Address="2001:0db8:85a3::8a2e:370:7334";ipv6SubnetP
refixLength=64;ipv6DefaultGateway="2001:db8:85a3::8a2e:370:ffff"}
```

```
Invoke-TSMCLI -CommandName set-nicsettings -ComputerName myserver -Port 80
-Authentication basic -Credential $credential -CommandArgs @
{nicMode="shared";ipv4Mode="dhcp"}
```

```
Invoke-TSMCLI -CommandName set-nicsettings -ComputerName myserver -Port 80
-Authentication basic -Credential $credential -CommandArgs @
{ipv4Mode="static";ipv4Address="192.168.1.12";ipv4SubnetMask="255.255.255.
0";ipv4DefaultGateway="192.168.1.1";ipv6Mode="static";ipv6Address="2001:0d
b8:85a3::8a2e:370:7334";ipv6SubnetPrefixLength=64;ipv6DefaultGateway="200
1:db8:85a3::8a2e:370:ffff"}
```

```
Invoke-TSMCLI -CommandName set-nicsettings -ComputerName myserver -Port 80
-Authentication basic -Credential $credential -CommandArgs @
{ipv4Mode="static";ipv4Address="192.168.2.12";ipv4SubnetMask="255.255.255.
0";ipv4DefaultGateway="192.168.2.1";vlanEnabled="1";vlanId="2";vlanPriorit
y="1"}
```

## RESPONSE

Object Result		
Integer	<b>Result</b>	The response code for "Success" (always 0).
String	<b>Message</b>	The localized message for "Success".

## set-ntpsettings

Sets the current network time server (NTP) configuration on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] set-ntpsettings -ComputerName <ComputerName>
[-Port <PortNumber>] [-Authentication <AuthenticationMode>] -Credential
<Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-CommandArgs] @
{ntpServer1="<Server Domain Name|IPv4 Address|IPv6
Address>"[;ntpServer2="<Server Domain Name|IPv4 Address|IPv6 Address>"]}
[-LogLevel <0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The set-ntpsettings command sets the network time server (NTP) configuration on a specific Lenovo ThinkServer System Manager. This configuration is comprised of a primary NTP server and an optional secondary NTP server.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### ntpServer1

Specifies the address for the primary Network Time Server. It can be a Server Domain Name, an IPv4 address or an IPv6 address.

#### ntpServer2

Specifies the address for the secondary Network Time Server (optional). It can be a Server Domain Name, an IPv4 address, an IPv6 address, or even an empty string ("").

### EXAMPLES

```
Invoke-TSMCLI -CommandName set-ntpsettings -ComputerName myserver -Port 80
-Authentication basic -Credential $cred -CommandArgs @
{ntpServer1="pool.ntp.org"}
```

```
Invoke-TSMCLI -CommandName set-ntpsettings -ComputerName myserver -Port 80
-Authentication basic -Credential $cred -CommandArgs @
{ntpServer1="pool.ntp.org"; ntpServer2="fd7d:fled:817e:52b:0:0:0:1"}
```

```
Invoke-TSMCLI -CommandName set-ntpsettings -ComputerName myserver -Port 80
-Authentication basic -Credential $cred -CommandArgs @
{ntpServer1="192.168.1.258"; ntpServer2="10.10.58.69"}
```

## RESPONSE

		Object Result
Integer	<b>Result</b>	The success response code. 0 (zero) when NTP configuration was set and the TSM was able to synchronize with the NTP servers. 1 when NTP configuration was set but the TSM was unable to synchronize with the NTP servers.
String	<b>Message</b>	The localized message for the scenarios described in the "Result" field.

## set-pamorder

Modifies the PAM ordering on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] set-pamorder -ComputerName <ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>] -Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-CommandArgs] @ {order="<pam_module>,<pam_module>,<pam_module>"} [-LogLevel <0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The set-pamorder command modifies the PAM ordering for user authentication on a specific Lenovo ThinkServer System Manager.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### order

A comma separated list with the PAM modules ordering (required). Valid values for each <pam\_module> are: "ipmi", "ldap" and "ad". It must not have two or more entries with the same value.

### EXAMPLES

```
Invoke-TSMCLI -CommandName set-pamorder -ComputerName myserver -Port 80 -Authentication basic -Credential $cred -CommandArgs @ {order="ipmi,ldap,ad"}
```

### RESPONSE

Object Result		
Integer	<b>Result</b>	The response code for "Success" (always 0).
String	<b>Message</b>	The localized message for "Success".



## set-powerbuttonstate

Sets the power button state (enabled or disabled) on a specific ThinkServer.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] set-powerbuttonstate -ComputerName
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]
-Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-
CommandArgs] @{powerButtonEnabled="<0|1>"} [-LogLevel <0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The set-powerbuttonstate command sets the power button state (enabled or disabled) on a specific ThinkServer. Note that when the power button is disabled it is still possible to use it to power on the ThinkServer.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### powerButtonEnabled

The power button state (0 for "Disabled" or 1 for "Enabled") (required).

### EXAMPLES

```
Invoke-TSMCLI -CommandName set-powerbuttonstate -ComputerName myserver -
Port 80 -Authentication basic -Credential $cred -CommandArgs @
{powerButtonEnabled="1"}
```

### RESPONSE

		Object Result
Integer	<b>Result</b>	The response code for "Success" (always 0).
String	<b>Message</b>	The localized message for "Success".

## set-powerconsumptionsettings

Sets the power consumption history configuration on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] set-powerconsumptionsettings -ComputerName
<ComputerName> [-port <PortNumber>] [-Authentication <AuthenticationMode>]
-Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-
CommandArgs] @{{days="<days>"}[;interval="<interval>"]} [-LogLevel <0|1|2|
3|4|5|6|7>]
```

### DESCRIPTION

The set-powerconsumptionsettings command configures the period in days and the interval in minutes of the power consumption history data that will be recorded on a specific Lenovo ThinkServer System Manager. At least one optional argument must be provided. Note that when these settings are changed the power consumption history is reset.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### days

The number of days the records will be kept by the TSM (optional). Valid values are numbers between 1 and 45 inclusive.

#### interval

The recording interval for each power consumption record, in minutes (optional). Valid values are numbers between 1 and 60 inclusive.

### EXAMPLES

```
Invoke-TSMCLI -CommandName set-powerconsumptionsettings -ComputerName
myserver -port 80 -Authentication basic -Credential $cred -CommandArgs @
{days="30";interval="60"}
```

### RESPONSE

Object Result		
Integer	<b>Result</b>	The response code for "Success" (always 0).
String	<b>Message</b>	The localized message for "Success".

## set-powerstate

Sets a new power state on a specific ThinkServer.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] set-powerstate -ComputerName <ComputerName>
[-Port <PortNumber>] [-Authentication <AuthenticationMode>] -Credential
<credential> [-UseSSL [-SkipSSLCertificateCheck]] [-CommandArgs] @
{operation=<"on"|"reset"|"off-immediate"|"off-graceful"|"powercycle">} [-
LogLevel <0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The set-powerstate command requests a power control operation on the ThinkServer managed by the specified Lenovo ThinkServer System Manager. This command returns success when the power control operation is sent to the TSM, but the actual operation is done asynchronously. To monitor the server power state, use the "get-powerstate" command.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### operation

Specifies the power state to be set on the ThinkServer. Possible values are: "on" to turn on the ThinkServer, "off-graceful" to initiate an operating system shutdown, "off-immediate" to immediately power off the ThinkServer, "reset" to reboot the ThinkServer without powering it off first, "powercycle" to first power off and then power on again the ThinkServer. The "off-graceful" operation will depend on operating system default behavior of power button, please advise operating system vendor to change the behavior.

### EXAMPLES

```
Invoke-TSMCLI -CommandName set-powerstate -ComputerName myserver -Port 80
-Authentication basic -Credential $credential -CommandArgs @
{operation="on"}
```

### RESPONSE

Object Result	
Integer	<b>Result</b> The response code for "Success" (always 0).
String	<b>Message</b> The localized message for "Success".

## set-psuredundancysettings

Configures the power supply redundancy on a specific ThinkServer.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] set-psuredundancysettings -ComputerName
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]
-Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-
CommandArgs] @{[redundancyMode="<normal|cold>"] [;powerCappingEnabled="<0|
1>"]}
```

### DESCRIPTION

The set-psuredundancysettings command configures the power supply redundancy on a specific ThinkServer.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### redundancyMode

The redundancy mode. Possible values are: "normal" or "cold" (optional and the new configuration will be active only when there is two PSUs active on the server). For more information about PSU redundancy modes please verify the ThinkServer documentation.

#### powerCappingEnabled

Specifies the power capping enable state (optional and the new configuration will be active only when there is two PSUs active on the server).

### EXAMPLES

```
Invoke-TSMCLI -CommandName set-psuredundancysettings -ComputerName
myserver -Port 80 -Authentication basic -Credential $cred -CommandArgs @
{redundancyMode="normal"}
```

```
Invoke-TSMCLI -CommandName set-psuredundancysettings -ComputerName
myserver -Port 80 -Authentication basic -Credential $cred -CommandArgs @
{powerCappingEnabled="1"}
```

### RESPONSE

Object Result	
Integer	<b>Result</b> The response code for "Success" (always 0).

String	<b>Message</b>	The localized message for "Success".
--------	----------------	--------------------------------------

## set-remotemediasettings

Sets the remote media settings on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] set-remotemediasettings -ComputerName
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]
-Credential <credential> [-UseSSL [-SkipSSLCertificateCheck]] [-
CommandArgs] @{enabled=0} [-LogLevel <0|1|2|3|4|5|6|7>]
```

```
Invoke-TSMCLI [-CommandName] set-remotemediasettings -ComputerName
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]
-Credential <credential> [-UseSSL [-SkipSSLCertificateCheck]] [-
CommandArgs] @{enabled=1; serverAddress="<server_address>";
sourcePath="<source_path>"; shareType="cifs"; username="<username>";
password="<password>"; domainName="<domain_name>"} [-LogLevel <0|1|2|3|4|
5|6|7>]
```

```
Invoke-TSMCLI [-CommandName] set-remotemediasettings -ComputerName
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]
-Credential <credential> [-UseSSL [-SkipSSLCertificateCheck]] [-
CommandArgs] @{enabled=1; serverAddress="<server_address>";
sourcePath="<source_path>"; shareType="nfs"} [-LogLevel <0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The set-remotemediasettings command sets the remote media settings on a specific Lenovo ThinkServer System Manager. This command allows enabling or disabling the remote media support and also configuring a server where remote images will be placed.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### enabled

Indicates if remote media support should be enabled (1) or not (0).

#### serverAddress

The remote media server address. It can be an IPv4 address or an IPv6 address.

#### sourcePath

Path where the image files are located on the specified server.

#### shareType

The type of share used on the specified server. Either "nfs" or "cifs".

**username**

If the share type is "cifs", then a username for the server must be provided. It shall have a maximum of 255 characters. Blank spaces at the beginning and the end of the username are discarded. The username cannot contain only blank spaces. The following special characters are not allowed: '#'(pound sign), ';' (semicolon), '\' (backslash) and '"' (single quote).

**password**

If the share type is "cifs", then the user's password for the server must be provided. It shall have a maximum of 31 characters.

**domainName**

If the share type is "cifs", then the domain name must be provided.

**EXAMPLES**

```
Invoke-TSMCLI -CommandName set-remotemediasettings -ComputerName myserver
-Port 80 -Credential $cred -CommandArgs @{enabled=1;
serverAddress="192.168.1.104"; sourcePath="media"; shareType="cifs";
username="user"; password="1234"; domainName="mydomain.localdomain"}
```

```
Invoke-TSMCLI -CommandName set-remotemediasettings -ComputerName myserver
-Port 80 -Credential $cred -CommandArgs @{enabled=1;
serverAddress="192.168.1.104"; sourcePath="media"; shareType="nfs"}
```

```
Invoke-TSMCLI -CommandName set-remotemediasettings -ComputerName myserver
-Port 80 -Credential $cred -CommandArgs @{enabled=0}
```

**RESPONSE**

Object Result	
Integer	<b>Result</b> The response code for "Success" (always 0).
String	<b>Message</b> The localized message for "Success".

## set-service

Updates the settings for the services available on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] set-service -ComputerName <ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>] -Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-CommandArgs] @ {serviceId="<service_id>" [;nonsecurePort="<port_number>"] [;securePort="<port_number>"] [;enabled="<0|1>"] [;force="<0|1>"]} [-LogLevel <0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The set-service command allows user to change secure and nonsecure port numbers of the services available on a specific Lenovo ThinkServer System Manager. It also allows to enable and disable the specified service. Please note that we should be carefully to set the following ports to either 'nonsecurePort' or 'securePort', because those ports are reserved for TSM internal usage: 22, 23, 80, 123, 161, 199, 427, 443, 546, 555, 623, 1900, 5120, 5122, 5123, 5124, 5126, 5127, 5988, 7578, 7582 and 50000. If those ports are occupied, it might cause TSM not working properly.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### serviceId

Specifies the id of the service of which settings will be updated. The serviceId can be obtained through get-services command.

#### nonsecurePort

Specifies a new port number to be used as the nonsecure port of the specified service. It shall be a value from 1 to 65535. If the service does not support nonsecure port, this argument should not be provided.

#### securePort

Specifies a new port number to be used as the secure port of the specified service. It shall be a value from 1 to 65535. If the service does not support secure port, this argument should not be provided.

#### enabled

Indicates if the service should be enabled (1) or not (0). Caution: TSMCLI relies on Web service to communicate with TSM. Disabling web service will stop communication between the TSMCLI and the target TSM.

#### force



As a security measure, the "force" argument is required when disabling the web service and must be specified as "1".

## EXAMPLES

```
Invoke-TSMCLI -CommandName set-service -ComputerName myserver -Port 80 -
Authentication basic -Credential $cred -CommandArgs @
{serviceId="32";securePort="22"}
```

```
Invoke-TSMCLI -CommandName set-service -ComputerName myserver -Port 80 -
Authentication basic -Credential $cred -CommandArgs @
{serviceId="64";nonsecurePort="23"}
```

```
Invoke-TSMCLI -CommandName set-service -ComputerName myserver -Port 80 -
Authentication basic -Credential $cred -CommandArgs @
{serviceId="1";nonsecurePort="80";securePort="443"; enabled="1"}
```

## RESPONSE

Object Result		
Integer	<b>Result</b>	The response code for "Success" (always 0).
String	<b>Message</b>	The localized message for "Success".

## set-smtpsettings

Modifies the SMTP settings on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] set-smtpsettings -ComputerName <ComputerName>
[-Port <PortNumber>] [-Authentication <AuthenticationMode>] -Credential
<Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-CommandArgs] @
{[senderAddress="<email address>"][;machineName="<machine name>"]
[;server1Enabled="<0|1>"][;server1Port="<1 to 65535>"]
[;server1Address="<ip address>"][;server1AuthEnabled="<0|
1>"][;server1Username="<username>;server1Password="<password>"]]
[;server2Enabled="<0|1>"][;server2Port="<1 to 65535>"]
[;server2Address="<ip address>"][;server2AuthEnabled="<0|
1>"][;server2Username="<username>;server2Password="<password>"]] } [-
LogLevel <0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The set-smtpsettings command modifies the SMTP settings on a specific Lenovo ThinkServer System Manager. This command will keep the current settings modifying only the fields provided on command arguments. At least one optional command argument must be provided. If either the SMTP server 1 or SMTP server 2 has authentication enabled, the AuthEnabled, Username and Password arguments must be provided for that server.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### senderAddress

The Sender Address on the SMTP server (optional). The Sender Address length is limited to 63 characters and it shall be in the format <username>@<domain>. Examples: "myname@mycompany.com" and "mymail@dept.example.com".

#### machineName

The Machine Name of the SMTP server (optional). It shall be at most 15 characters long and can contain only letters and digits.

#### server1Enabled

Indicates if the SMTP server 1 should be enabled (1) or not (0) (optional).

#### server1Port

The SMTP port for the SMTP server 1 (optional). It is a number from 1 to 65535 where default is 25 if not priorly defined.

**server1Address**

The IP address of the SMTP server 1 (optional). It can be an IPv4 or an IPv6 address format.

**server1AuthEnabled**

Indicates if the SMTP server 1 requires authentication (1) or not (0) (optional).

**server1Username**

The username to access the SMTP server 1 account (required when server1AuthEnabled is 1). It shall be at least 4 characters long and at most 64 characters long and must start with an alphabetical character. The following characters are allowed: [a-zA-Z] (English alphabet with uppercase and lowercase form) and [0-9] (number from 0 to 9), along with dash and underscore.

**server1Password**

The password of the SMTP server 1 account (required when server1AuthEnabled is 1). It shall be at least 4 characters long and at most 64 characters long. White spaces are not allowed.

**server2Enabled**

Indicates if the SMTP server 2 should be enabled (1) or not (0) (optional).

**server2Port**

The SMTP port for the SMTP server 2 (optional). It is a number from 1 to 65535 where default is 25 if not priorly defined.

**server2Address**

The IP address of the SMTP server 2 (optional). It can be an IPv4 or an IPv6 address format.

**server2AuthEnabled**

Indicates if the SMTP server 2 requires authentication (1) or not (0) (optional).

**server2Username**

The username to access the SMTP server 2 account (required when server2AuthEnabled is 1). It shall be at least 4 characters long and at most 64 characters long and must start with an alphabetical character. The following characters are allowed: [a-zA-Z] (English alphabet with uppercase and lowercase form) and [0-9] (number from 0 to 9), along with dash and underscore.

**server2Password**

The password of the SMTP server 2 account (required when server2AuthEnabled is 1). It shall be at least 4 characters long and at most 64 characters long. White spaces are not allowed.

**EXAMPLES**

```
Invoke-TSMCLI -CommandName set-smtpsettings -ComputerName myserver -Port  
80 -Authentication basic -Credential $cred -CommandArgs @  
{server1Enabled="1";server1Port="25";server1Address="192.168.1.130";server
```

```
1AuthEnabled="1";server1Username="username";server1Password="1234";server2  
Enabled="0" }
```

## RESPONSE

Object Result		
Integer	<b>Result</b>	The response code for "Success" (always 0).
String	<b>Message</b>	The localized message for "Success".

## set-solsettings

Sets the serial over LAN settings on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] set-solsettings -ComputerName <ComputerName>
[-Port <PortNumber>] [-Authentication <AuthenticationMode>] -Credential
<Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-CommandArgs] @
{solEnabled="0"} [-LogLevel <0|1|2|3|4|5|6|7>]
```

```
Invoke-TSMCLI [-CommandName] set-solsettings -ComputerName <ComputerName>
[-Port <PortNumber>] [-Authentication <AuthenticationMode>] -Credential
<Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-CommandArgs] @
{[solEnabled="1";][baudRate="<9600|19200|38400|57600|115200>";]
[flowControl="<none|xonxoff|hw>";][dtrHangUp="<0|1>"]} [-LogLevel <0|1|2|
3|4|5|6|7>]
```

### DESCRIPTION

The set-solsettings command sets the serial over LAN (SOL) settings on a specific Lenovo ThinkServer System Manager.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### solEnabled

Indicates if the SOL should be enabled (1) or disabled (0). (required if enabling or disabling SOL)

#### baudRate

The baud rate. Valid values are "9600", "19200", "38400", "57600" and "115200". (required if solEnabled is changed from "0" to "1")

#### flowControl

The flow control configuration. Valid values are "none" for no flow control, "xonxoff" for software flow control and "hw" for hardware flow control. (required if solEnabled is changed from "0" to "1")

#### dtrHangUp

Indicates if the DTR hang up should be enabled (1) or disabled (0). (required if solEnabled is changed from "0" to "1")

## EXAMPLES

```
Invoke-TSMCLI -CommandName set-solsettings -ComputerName myserver -Port 80
-Authentication basic -Credential $cred -CommandArgs @{solEnabled="0"}
```

```
Invoke-TSMCLI -CommandName set-solsettings -ComputerName myserver -Port 80
-Authentication basic -Credential $cred -CommandArgs @
{solEnabled="1";baudRate="57600";flowControl="xonxoff";dtrHangUp="0"}
```

```
Invoke-TSMCLI -CommandName set-solsettings -ComputerName myserver -Port 80
-Authentication basic -Credential $cred -CommandArgs @{baudRate="57600"}
```

## RESPONSE

Object Result		
Integer	<b>Result</b>	The response code for "Success" (always 0).
String	<b>Message</b>	The localized message for "Success".

## set-sslcertificate

Replaces the SSL certificate on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] set-sslcertificate -ComputerName
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]
-Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-
CommandArgs] @
{certificateFile="<path_to_certificate_file>";privateKeyFile="<path_to_pri
vate_key_file>"} [-LogLevel <0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The set-sslcertificate command uploads SSL certificate and private key files to a specific Lenovo ThinkServer System Manager, replacing the previous ones. The TSM SSL certificate is used by clients to verify the identity of the TSM they are connecting to.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### certificateFile

The SSL certificate file to upload to the TSM (it must have a ".pem" extension).

#### privateKeyFile

The private key file to upload to the TSM (it must have a ".pem" extension).

### EXAMPLES

```
Invoke-TSMCLI -CommandName set-sslcertificate -ComputerName myserver -Port
80 -Authentication basic -Credential $cred -CommandArgs @
{certificateFile="C:\MyCertFolder\tsm_cert.pem";privateKeyFile="C:
\MyCertFolder\tsm_key.pem"}
```

### RESPONSE

Object Result		
Integer	<b>Result</b>	The response code for "Success" (always 0).
String	<b>Message</b>	The localized message for "Success".

## set-timezone

Sets the timezone of a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] set-timezone -ComputerName <ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>] -Credential <credential> [-UseSSL [-SkipSSLCertificateCheck]] [-CommandArgs] @ {timezone="<timezone_id>"} [-LogLevel <0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The set-timezone command sets the timezone of a specific Lenovo ThinkServer System Manager.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### timezone

Specifies the timezone identifier. To get the list of valid timezones identifiers on the TSM, use the command "get-timezones".

### EXAMPLES

```
Invoke-TSMCLI -CommandName set-timezone -ComputerName myserver -Port 80 -Authentication basic -Credential $credential -CommandArgs @ {timezone="America/Sao_Paulo"}
```

### RESPONSE

Object Result		
Integer	<b>Result</b>	The response code for "Success" (always 0).
String	<b>Message</b>	The localized message for "Success".



## set-user

Modifies an user account on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] set-user -ComputerName <ComputerName> [-Port
<PortNumber>] [-Authentication <AuthenticationMode>] -Credential
<credential> [-UseSSL [-SkipSSLCertificateCheck]] [-CommandArgs] @
{userId="<user_id>" [;password="<password>"] [;role="<admin|user|operator>"]
[;email="<email>"] [;kvm="<0|1>"] [;virtualMedia="<0|1>"] [[;snmp="<0|1>"]
[;snmpAccess="<read-only|read-write>"] [;authenticationProtocol="<sha|
md5>"] [;privacyProtocol="<des|aes>"]]} [-LogLevel <0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The set-user command modifies an user account on a specific Lenovo ThinkServer System Manager.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### userId

Specifies the user id.

#### password

Specifies the password (required if "authenticationProtocol" argument is specified). It shall have a maximum of 20 characters and only printable ASCII characters and spaces are accepted. In addition, if SNMP is enabled, the password shall have at least 8 characters.

#### role

Specifies the user role (optional). Possible values are: "admin" for Administrator, "user" for User or "operator" for Operator.

#### email

Specifies the user email (optional). The email length is limited to 63 characters and can be defined as empty string ("") to clear the former value. In addition, the e-mail shall be in the format <username>@<domain>. Examples: "myname@mycompany.com" and "my.mail@dept.example.com".

#### kvm

Specifies the KVM privilege (optional).

#### virtualMedia

Specifies the Virtual Media privilege (optional).

**snmp**

Specifies the SNMP enabled state (optional).

**snmpAccess**

Specifies the SNMP access level (required if SNMP is changed from "0" to "1"). Possible values are: "read-only" or "read-write".

**authenticationProtocol**

Specifies the authentication protocol for SNMP settings (required if SNMP is changed from "0" to "1"). Possible values are: "sha" or "md5". The "password" argument is required if authentication protocol is specified.

**privacyProtocol**

Specifies the encryption algorithm to use for SNMP settings (required if SNMP is changed from "0" to "1"). Possible values are: "des" or "aes".

**EXAMPLES**

```
Invoke-TSMCLI -CommandName set-user -ComputerName myserver -Port 80 -
Authentication basic -Credential $credential -CommandArgs @
{userId=5;password="newpass"}
```

```
Invoke-TSMCLI -CommandName set-user -ComputerName myserver -Port 80 -
Authentication basic -Credential $credential -CommandArgs @
{userId=3;role="user";email="newemail@domain.com"}
```

```
Invoke-TSMCLI -CommandName set-user -ComputerName myserver -Port 80 -
Authentication basic -Credential $credential -CommandArgs @
{userId=6;password="1234";role="operator";email="newemail@domain.com";kvm=
"1";snmp="1";snmpAccess="read-
only";authenticationProtocol="md5";privacyProtocol="aes"}
```

**RESPONSE**

Object Result		
Integer	<b>Result</b>	The response code for "Success" (always 0).
String	<b>Message</b>	The localized message for "Success".

## set-websessiontimeout

Modifies the TSM Web UI session timeout on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] set-websessiontimeout -ComputerName
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]
-Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-
CommandArgs] @{timeout="<timeout_in_seconds>"} [-LogLevel <0|1|2|3|4|5|6|
7>]
```

### DESCRIPTION

The set-websessiontimeout command modifies the TSM Web UI session timeout on a specific Lenovo ThinkServer System Manager.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### timeout

The new value (in seconds) for the Web session timeout (required). Valid values are between 300 and 1800, in multiples of 300.

### EXAMPLES

```
Invoke-TSMCLI -CommandName set-websessiontimeout -ComputerName myserver -
Port 80 -Authentication basic -Credential $cred -CommandArgs @
{timeout="1500"}
```

### RESPONSE

Object Result		
Integer	<b>Result</b>	The response code for "Success" (always 0).
String	<b>Message</b>	The localized message for "Success".

## start-fwupdate

Starts the firmware update on a specific ThinkServer.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] start-fwupdate -ComputerName <ComputerName>
[-Port <PortNumber>] [-Authentication <AuthenticationMode>] -Credential
<Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-CommandArgs] @
{updateId="<update_id>;rebootAfterUpdate="auto|manual"} [-LogLevel <0|1|
2|3|4|5|6|7>]
```

### DESCRIPTION

The start-fwupdate command starts the firmware update of components on a specific ThinkServer. The update can be performed only when the available firmware version is higher than the installed firmware version. The list of components and information about installed and available versions can be verified with the get-fwversions TSMCLI command.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### updateId

Specifies the update mode identifier (required).

#### rebootAfterUpdate

Defines the behavior for automatic reboots for some firmware update operations (required). Possible values are: "auto" or "manual". The update process for some components involves a TSM or Host reboot. In some of these cases, the reboot is needed during the update operation, and cannot be avoided. In some other cases, however, even though the update will take place only after rebooting, the reboot can be deferred by specifying the "manual" option, and will not happen automatically. Later on, when a reboot happens, the update is automatically applied. To automatically start the reboot, specify the "auto" option. The reboot requirements for a component update can be verified with the "get-fwversions" command.

### EXAMPLES

```
Invoke-TSMCLI -CommandName start-fwupdate -ComputerName myserver -Port 80
-Authentication basic -Credential $cred -CommandArgs @
{updateId="3199709292";rebootAfterUpdate="manual"}
```

```
Invoke-TSMCLI -CommandName start-fwupdate -ComputerName myserver -Port 80
-Authentication basic -Credential $cred -CommandArgs @
{updateId="3199709292";rebootAfterUpdate="auto"}
```

## RESPONSE

Object Result		
Integer	<b>Result</b>	The response code for Success (always 0).
String	<b>Message</b>	The localized message for Success.

## start-remotemediainage

Starts a remote media image redirection on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] start-remotemediainage -ComputerName
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]
-Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-
CommandArgs] @{imageType="<cddvd|hd>"; imageName="<image_name>"} [-
LogLevel <0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The start-remotemediainage command starts a remote media image redirection on a specific Lenovo ThinkServer System Manager. This operation can be executed only when remote media is enabled.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### imageType

Specifies the type of the remote image redirection to be started on the specified TSM. The possible values are "cddvd" for CD/DVD or "hd" for Harddisk image.

#### imageName

The name and format of the image file that will be accessed remotely from the TSM. The image file format must always be .img for Harddisk and .iso for CD/DVD. The image name is case sensitive.

### EXAMPLES

```
Invoke-TSMCLI -CommandName start-remotemediainage -ComputerName myserver -
Port 80 -Authentication basic -Credential $cred -CommandArgs @
{imageType="hd"; imageName="hdimage.img"}
```

### RESPONSE

Object Result		
Integer	<b>Result</b>	The response code for "Success" (always 0).
String	<b>Message</b>	The localized message for "Success".

## stop-pendingupdates

Stops all pending updates on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] stop-pendingupdates -ComputerName
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]
-Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-
CommandArgs] @{updateId="<update_id>";force="1"} [-LogLevel <0|1|2|3|4|5|
6|7>]
```

### DESCRIPTION

The stop-pendingupdates command stops all pending updates with pending status on a specific Lenovo ThinkServer System Manager.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### updateId

Specifies the update mode identifier (required).

#### force

The stop-pendingupdates command will cancel all pending updates. As a security measure, the "force" argument is required and must be specified as "1".

### EXAMPLES

```
Invoke-TSMCLI -CommandName stop-pendingupdates -ComputerName myserver -
Port 80 -Authentication basic -Credential $cred -CommandArgs @
{updateId="3199709292";force="1"}
```

### RESPONSE

		Object Result
Integer	<b>Result</b>	The response code for "Success" (always 0).
String	<b>Message</b>	The localized message for "Success".

## stop-remotemediainage

Stops a remote media image redirection on a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] stop-remotemediainage -ComputerName
<ComputerName> [-Port <PortNumber>] [-Authentication <AuthenticationMode>]
-Credential <Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-
CommandArgs] @{imageType="<cddvd|hd>"} [-LogLevel <0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The stop-remotemediainage command stops a remote media image redirection on a specific Lenovo ThinkServer System Manager. This operation can be executed only when remote media is enabled.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### imageType

Specifies the type of the remote image redirection to be stopped on the specified TSM. The possible values are "cddvd" for CD/DVD or "hd" for hard disk image.

### EXAMPLES

```
Invoke-TSMCLI -CommandName stop-remotemediainage -ComputerName myserver -
Port 80 -Authentication basic -Credential $cred -CommandArgs @
{imageType="hd"}
```

### RESPONSE

Object Result		
Integer	<b>Result</b>	The response code for "Success" (always 0).
String	<b>Message</b>	The localized message for "Success".



## transfer-fwimage

Transfers a firmware image to a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] transfer-fwimage -ComputerName <ComputerName>
[-Port <PortNumber>] [-Authentication <AuthenticationMode>] -Credential
<Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-CommandArgs] @
{updateId="<update_id>";serverAddress="<server_ip_address>";shareType="<nfs|
cifs|
tftp>";filename="<filename>"[;sourcePath="<source_path>";username="<userna
me>";password="<password>"]} [-LogLevel <0|1|2|3|4|5|6|7>]
```

### DESCRIPTION

The transfer-fwimage command transfers a firmware image located on a network location to a specific Lenovo ThinkServer System Manager. This image will be used to update the ThinkServer components.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### updateId

Specifies the update mode identifier (required).

#### serverAddress

The remote media server address. It can be an IPv4 address or an IPv6 address (required).

#### shareType

The share type used on the specified server (required). Valid values are "nfs", "cifs" or "tftp".

#### filename

The name of the image file that will be transferred to the TSM (required).

#### sourcePath

Path where the image files are located on the specified server (only required when shareType is "nfs" or "cifs").

#### username

The username for the server (only required when shareType is "cifs"). It shall have a maximum of 16 characters, must start with an alphabetical character and is case sensitive. The following special characters are not allowed: \[ ] ; | = , + \* ? < > @ # % " ' .

#### password

The user's password for the server (only required when shareType is "cifs"). All ASCII characters are accepted, except space.

## EXAMPLES

```
Invoke-TSMCLI -CommandName transfer-fwimage -ComputerName myserver -Port
80 -Authentication basic -Credential $cred -CommandArgs @
{updateId="3199709292";serverAddress="192.168.1.104";sourcePath="firmwareI
images";shareType="cifs";username="myuser";password="1234";filename="myFirm
wareFile"}
```

```
Invoke-TSMCLI -CommandName transfer-fwimage -ComputerName myserver -Port
80 -Authentication basic -Credential $cred -CommandArgs @
{updateId="3199709292";serverAddress="192.168.1.104";sourcePath="\firmware
Images";shareType="nfs";filename="myFirmwareFile"}
```

```
Invoke-TSMCLI -CommandName transfer-fwimage -ComputerName myserver -Port
80 -Authentication basic -Credential $cred -CommandArgs @
{updateId="3199709292";serverAddress="192.168.1.104";shareType="tftp";file
name="myFirmwareFile"}
```

## RESPONSE

Object Result		
Integer	<b>Result</b>	The response code for "Success" (always 0).
String	<b>Message</b>	The localized message for "Success".

## upload-fwimage

Uploads a firmware image to a specific TSM.

### SYNTAX

```
Invoke-TSMCLI [-CommandName] upload-fwimage -ComputerName <ComputerName>
[-Port <PortNumber>] [-Authentication <AuthenticationMode>] -Credential
<Credentials> [-UseSSL [-SkipSSLCertificateCheck]] [-CommandArgs] @
{updateId="<update_id>";fwFile="<path_to_image_file>"} [-LogLevel <0|1|2|
3|4|5|6|7>]
```

### DESCRIPTION

The upload-fwimage command uploads a firmware image file to a specific Lenovo ThinkServer System Manager. This image will be used to update the ThinkServer components.

### PARAMETERS

For a list of parameters applicable to all TSMCLI commands see [Chapter 5, TSMCLI Usage](#).

### COMMAND ARGUMENTS

#### updateId

Specifies the update mode identifier (required).

#### fwFile

The firmware image file to upload to the TSM (required).

### EXAMPLES

```
Invoke-TSMCLI -CommandName upload-fwimage -ComputerName myserver -Port 80
-Authentication basic -Credential $cred -CommandArgs @
{updateId="3199709292";fwFile="C:\MyFirmwareImageFolder\myFirmwareFile"}
```

### RESPONSE

Object Result	
Integer	<b>Result</b> The response code for "Success" (always 0).
String	<b>Message</b> The localized message for "Success".

## 8. Commands permissions

The table below shows the minimum permission for a TSM user required to successfully run each TSMCLI command:

Command	Minimum Privilege
backup-config	Administrator
clear-sel	Operator
create-adrolegroup	Administrator
create-alertnotification	Administrator
create-alertpolicy	Administrator
create-credential	User
create-eventfilter	Administrator
create-firewalliprule	Administrator
create-firewallportrule	Administrator
create-landestination	Administrator
create-ldaprolegroup	Administrator
create-user	Administrator
delete-adrolegroup	Administrator
delete-alertpolicy	Administrator
delete-eventfilter	Administrator
delete-firewalliprule	Administrator
delete-firewallportrule	Administrator
delete-landestination	Administrator
delete-ldaprolegroup	Administrator
delete-user	Administrator
enter-fwupdatemode	Administrator
exit-fwupdatemode	Administrator
firmware-update	Administrator
get-adrolegroups	Operator
get-adsettings	Operator
get-alertpolicies	Operator
get-auditlog	Operator
get-auditlogsettings	User
get-commands	-
get-crashscreen	User
get-datetime	User

get-dns	Operator
get-eventfilters	Operator
get-firewallrules	User
get-frus	User
get-fwupdatestatus	Administrator
get-fwversions	Administrator
get-help	-
get-identifierstate	User
get-landestinations	Operator
get-ldaprolegroups	Operator
get-ldapsettings	Operator
get-niclinksettings	Operator
get-nics	Operator
get-nicsettings	Operator
get-ntpsettings	Operator
get-pamorder	User
get-powerbuttonstate	User
get-powerconsumption	User
get-powerconsumptionsettings	User
get-powerstate	User
get-psuredundancysettings	User
get-remotemediasettings	User
get-sel	User
get-sensors	User
get-sensortypes	User
get-services	User
get-smtpsettings	Operator
get-solsettings	Operator
get-sslcertificate	User
get-systeminfo	User
get-timezones	User
get-users	Operator
get-websessiontimeout	User
reset-config	Administrator
reset-powerconsumptionhistory	Administrator
restore-config	Administrator
set-adrolegroup	Administrator

set-adsettings	Administrator
set-alertpolicy	Administrator
set-auditlogsettings	Administrator
set-datetime	Administrator
set-dns	Administrator
set-eventfilter	Administrator
set-identifierstate	Administrator
set-landestination	Administrator
set-ldaprolegroup	Administrator
set-ldapsettings	Administrator
set-niclinksettings	Administrator
set-nicsettings	Administrator
set-ntpsettings	Administrator
set-pamorder	Administrator
set-powerbuttonstate	Administrator
set-powerconsumptionsettings	Administrator
set-powerstate	Operator
set-psuredundancyssettings	Administrator
set-remotemediasettings	Administrator
set-service	Administrator
set-smtpsettings	Administrator
set-solsettings	Administrator
set-sslcertificate	Administrator
set-timezone	Administrator
set-user	Administrator
set-websessiontimeout	Administrator
start-fwupdate	Administrator
start-remotemediainage	User
stop-pendingupdates	Administrator
stop-remotemediainage	User
transfer-fwimage	Administrator
upload-fwimage	Administrator

## 9. Ports Usage

The table below shows the ports usage list for TSM.

Port	Owner Module	Usage
80	Web server (webgo/lighttpd)	Listening for network connections on HTTP
443	Web server (webgo/lighttpd)	Listening for secured network connections on HTTPS
22	Secure Shell (sshd)	Secure SMASH-Lite session
23	Telnet	Telnet session
5120	CD/DVD media	To accept regular CD media redirection connections
5124	CD/DVD media	To accept secure (SSL based) CD media redirection connections
5122	Floppy media	To accept regular FD media redirection connections
5126	Floppy media	To accept secure (SSL based) FD media redirection connections
5123	HD media	To accept regular HD media redirection connections
5127	HD media	To accept secure (SSL based) HD media redirection connections
7578	KVM server (adviser)	To accept regular KVM redirection connections
7582	KVM server (adviser)	To accept secure (SSL based) KVM redirection connections
623	IPMI Over LAN	Lan Interface
1900	uPnP discovery	Used for uPnP based BMC discovery
50000	uPnP discovery	Used for uPnP based BMC discovery
555	WSMAN	Eventing daemon's listening port (Implemented not enabled)
5988	SFCB (WSMAN)	WSMAN related
427	SLPD	Service Locator
123	NTP	Network Time Protocol (NTP) - used for time synchronization (UDP Connection)
161	SNMP	SNMP listens on this port for incoming SNMP requests (UDP)
199	SNMP	SNMP listens on this port for incoming connect requests (from the SMUX peers and various otherTCP end-points connected to SMUX peers to exchange SMUX PDUs)
546	DHCPv6	DHCPv6 clients listen for DHCP messages on this port (UDP)

## 10. ThinkServer firmware update flow

The list below describes the necessary steps to perform a firmware update on a ThinkServer using TSMCLI.

1. Enter firmware update mode
  - This task is accomplished by running the `enter-fwupdatemode` TSMCLI command.
  - It will provide an update ID that should be informed on subsequent firmware update steps.
  - A proper error message will be displayed for the case there is another firmware update session alive or if there is a firmware update in progress.
2. Transfer a firmware bundle to the TSM
  - This task is accomplished by running the `upload-fwimage` or the `transfer-fwimage` TSMCLI commands.
  - For these commands you should inform the update ID to proceed.
  - A proper error message will be displayed for the case the update ID is invalid or expired or if there is a firmware update in progress.
3. Check the firmware version on the bundle and on the BMC
  - This task is accomplished by running the `get-fwversions` TSMCLI command.
  - For the devices with no firmware present on bundle, the TSMCLI will display the New Image Version as not available.
4. Start the firmware update
  - This task is accomplished by running the `start-fwupdate` TSMCLI command.
  - For this command you should inform the update ID to proceed.
  - A proper error message will be displayed for the case the update ID is invalid or expired or if there is a firmware update in progress.
5. Check the firmware update status
  - This task can be accomplished by running the `get-fwupdatestatus` TSMCLI command.
6. Stop pending updates (alternative step)
  - This task is accomplished by running the `stop-pendingupdates` TSMCLI command.
  - For this command you should inform the update ID to proceed.
  - Note that this command sends the cancel request to the TSMCLI. Only when the update status is pending, the updates can be cancelled.
  - A proper error message will be displayed for the case the update ID is invalid or expired.
7. Exit firmware update mode
  - This task is accomplished by running the `exit-fwupdatemode` TSMCLI command.
  - For this command you should inform the update ID to proceed.



- Note that this step is not required in case the TSM is rebooted.
- A proper error message will be displayed for the case the update ID is invalid or expired.

## 11. TSMCLI Exceptions and Errors

Below is the list of common errors and exceptions that can be thrown by the TSMCLI:

- **Memory Allocation Exceptions**  
Thrown in case of exceptions when allocating memory for internal TSMCLI operations.
- **Missing Parameters**  
Thrown when a required parameter for a TSMCLI command is missing.
- **Missing Arguments**  
Thrown when a required argument specific to a TSMCLI command is missing.
- **Invalid Arguments**  
Thrown when an invalid argument is informed on a TSMCLI command.
- **Invalid Argument Values**  
Thrown when invalid values are informed on a TSMCLI command specific argument.
- **Invalid Commands**  
Thrown when an invalid TSMCLI command is informed.
- **Server Communication Problems**  
Thrown when the TSM is not reachable or takes too long to respond and similar situations.
- **Authentication Exceptions**  
Thrown when the some authentication problem is found, like invalid or missing credentials.

## 12. Tips

### 12.1. How to enable alert notifications for the TSM using TSMCLI

The TSM is capable of sending SNMP traps and/or email alerts to notify users whenever a sensor exceeds a preset threshold value (e.g. the fan speed is too low). The following steps describe how to properly create an alert notification.

#### Step 1: Create a LAN Destination

By creating a LAN Destination, the TSM will be able to determine where to send the notification. It can be either a SNMP trap or an email.

**In order to create a SNMP trap LAN Destination, the following command can be used:**

```
Invoke-TSMCLI -CommandName create-landestination -ComputerName <TSM_IP> -
Credential <TSM_CREDENTIAL> -CommandArgs @{type="snmp";
destinationAddress="<SNMP_TRAP_RECEIVER_IP>"}
```

**In order to create an email LAN Destination, the following command can be used:**

```
Invoke-TSMCLI -CommandName create-landestination -ComputerName <TSM_IP> -
Credential <TSM_CREDENTIAL> -CommandArgs @{type="email";
userId="<TSM_USER_ID>"}
```

Make sure the selected user has an email address set. Refer to the [create-user](#) TSMCLI command for details.

After creating the new LAN Destination, TSMCLI will return a **LANDestinationId** that will be required in Step 2.

#### Step 2: Create an Alert Policy

An Alert Policy is used to provide a policy/action for a sensor related event on the TSM.

**In order to create an Alert Policy, the following command can be used:**

```
Invoke-TSMCLI -CommandName create-alertpolicy -ComputerName <TSM_IP> -
Credential <TSM_CREDENTIAL> -CommandArgs @
{policyNumber="<ALERT_POLICY_NUMBER>"; policyEnabled="1"; policySet="1";
lanDestinationId="<LANDestinationId>"; eventSpecificString="1";
alertStringKey="15"}
```

After creating the new Alert Policy, TSMCLI will return an **AlertPolicyId**.

On Step 3, the **ALERT\_POLICY\_NUMBER** set on this step will be required. Note that the policy number can vary from 1 to 15 and multiple **AlertPolicyIds** may be set to a single policy number. This way a single notification can be sent to multiple destinations.

### Step 3: Create an Event Filter

An Event Filter is used by the TSM to filter the sensor related events that need attention.

**In general, a good start is an event filter that watches all events that happened on a specified sensor. For this purpose the following command can be used:**

```
Invoke-TSMCLI -CommandName create-eventfilter -ComputerName <TSM_IP> -
Credential <TSM_CREDENTIAL> -CommandArgs @{eventFilterEnabled="1";
severity="noncritical"; powerAction="none";
policyNumber="<ALERT_POLICY_NUMBER>"; generatorIdRawData="1";
generatorId1="0xFF"; generatorId2="0xFF"; sensorMode="sensorname";
sensorName="<SENSOR_NAME>"; eventOptions="all"; eventTrigger="255";
eventData1ANDMask="255"; eventData1Compare1="255";
eventData1Compare2="255"; eventData2ANDMask="255";
eventData2Compare1="255"; eventData2Compare2="255";
eventData3ANDMask="255"; eventData3Compare1="255";
eventData3Compare2="255" }
```

After these steps the TSM will be ready to send SNMP traps and/or email notifications in case an abnormal situation is detected by a sensor.

## 12.2. Unsupported special characters on the TSM and TSMCLI

The TSM and, consequently, TSMCLI don't allow some special characters to be input as an argument. This section describes all limitations for each case.

- All ASCII control characters are not supported. See the table below:

DEC	OCT	HEX	Symbol	Description
0	000	00	NUL	Null character
1	001	01	SOH	Start of Heading
2	002	02	STX	Start of Text
3	003	03	ETX	End of Text
4	004	04	EOT	End of Transmission
5	005	05	ENQ	Enquiry
6	006	06	ACK	Acknowledgment
7	007	07	BEL	Bell
8	010	08	BS	Back Space
9	011	09	HT	Horizontal Tab
10	012	0A	LF	Line Feed
11	013	0B	VT	Vertical Tab
12	014	0C	FF	Form Feed
13	015	0D	CR	Carriage Return
14	016	0E	SO	Shift Out / X-On
15	017	0F	SI	Shift In / X-Off

DEC	OCT	HEX	Symbol	Description
16	020	10	DLE	Data Line Escape
17	021	11	DC1	Device Control 1 (oft. XON)
18	022	12	DC2	Device Control 2
19	023	13	DC3	Device Control 3 (oft. XOFF)
20	024	14	DC4	Device Control 4
21	025	15	NAK	Negative Acknowledgement
22	026	16	SYN	Synchronous Idle
23	027	17	ETB	End of Transmit Block
24	030	18	CAN	Cancel
25	031	19	EM	End of Medium
26	032	1A	SUB	Substitute
27	033	1B	ESC	Escape
28	034	1C	FS	File Separator
29	035	1D	GS	Group Separator
30	036	1E	RS	Record Separator
31	037	1F	US	Unit Separator

- For TSM Users, the following rules applies to "usernames":

- The username shall not start with a digit.
- The username shall not contain the following characters:

DEC	OCT	HEX	Symbol	Description
32	040	20		Space
33	041	21	!	Exclamation mark
34	042	22	"	Double quotes
35	043	23	#	Hash
36	044	24	\$	Dollar sign
37	045	25	%	Percent
38	046	26	&	Ampersand
39	047	27	'	Single quote
40	050	28	(	Opening parenthesis
41	051	29	)	Closing parenthesis
42	052	2A	*	Asterisk
43	053	2B	+	Plus
44	054	2C	,	Comma
46	056	2E	.	Period
47	057	2F	/	Slash
58	072	3A	:	Colon

DEC	OCT	HEX	Symbol	Description
59	073	3B	;	Semicolon
60	074	3C	<	Less than
61	075	3D	=	Equals
62	076	3E	>	Greater than
63	077	3F	?	Question mark
91	133	5B	[	Opening bracket
92	134	5C	\	Backslash
93	135	5D	]	Closing Bracket
94	136	5E	^	Caret
96	140	60	`	Grave accent
123	173	7B	{	Opening braces
124	174	7C		Vertical bar
125	175	7D	}	Closing braces
126	176	7E	~	Tilde
127	177	7F		Delete

- The username shall not start with any of the following characters:

DEC	OCT	HEX	Symbol	Description
45	055	2D	-	Hyphen
64	100	40	@	At symbol
95	137	5F	_	Underscore

- For TSM Users, the following characters are not allowed in "password":

DEC	OCT	HEX	Symbol	Description
127	177	7F		Delete

- For SMTP settings, the following characters are not allowed in "sender address":

DEC	OCT	HEX	Symbol	Description
39	047	27	'	Single quote
92	134	5C	\	Backslash
127	177	7F		Delete

- For SMTP settings, the following characters are not allowed in "server username":

1. The "server username" shall not start with a digit.
2. The "server username" shall not contain the following characters:

DEC	OCT	HEX	Symbol	Description
32	040	20		Space
33	041	21	!	Exclamation mark
34	042	22	"	Double quotes
35	043	23	#	Hash
36	044	24	\$	Dollar sign
37	045	25	%	Percent
38	046	26	&	Ampersand
39	047	27	'	Single quote
40	050	28	(	Opening parenthesis
41	051	29	)	Closing parenthesis
42	052	2A	*	Asterisk
43	053	2B	+	Plus
44	054	2C	,	Comma
45	055	2D	-	Dash
46	056	2E	.	Period
47	057	2F	/	Slash

DEC	OCT	HEX	Symbol	Description
58	072	3A	:	Colon
59	073	3B	;	Semicolon
60	074	3C	<	Less than
61	075	3D	=	Equals
62	076	3E	>	Greater than
63	077	3F	?	Question mark
64	100	40	@	At symbol
91	133	5B	[	Opening bracket
92	134	5C	\	Backslash
93	135	5D	]	Closing Bracket
94	136	5E	^	Caret
95	137	5F	_	Underscore
96	140	60	`	Grave accent
123	173	7B	{	Opening braces
124	174	7C		Vertical bar
125	175	7D	}	Closing braces
126	176	7E	~	Tilde

3. The "server username" shall not start with any of the following characters:

DEC	OCT	HEX	Symbol	Description
45	055	2D	-	Hyphen
95	137	5F	_	Underscore

- For SMTP settings, the following characters are not allowed in "password":

DEC	OCT	HEX	Symbol	Description
32	40	20		Space

- For Active Directory settings, the following characters are not allowed in "secret user":

DEC	OCT	HEX	Symbol	Description
34	042	22	"	Double quotes
42	052	2A	*	Asterisk
43	053	2B	+	Plus
44	054	2C	,	Comma
47	057	2F	/	Slash
58	072	3A	:	Colon
59	073	3B	;	Semicolon
60	074	3C	<	Less than
61	075	3D	=	Equals
62	076	3E	>	Greater than
63	077	3F	?	Question mark
64	100	40	@	At symbol
91	133	5B	[	Opening bracket
92	134	5C	\	Backslash
93	135	5D	]	Closing Bracket
124	174	7C		Vertical bar