Class GbtDefinition

```
java.lang.Object
    |
    +--Gbt.utility.API.GbtDefinition
```

< Constructors >

public class **GbtDefinition**
extends java.lang.Object

GbtUtility command line arguments definition of setting different command line functions for {@link GbtUtilityAPI} usage. This definition class including "rackst", "rackex", "loaddf", "update", "userapp", "vmcli" and "bmcutil" commands arguments setting methods for {@link GbtUtilityAPI} callable. The detail usage you can see more in each command line {@link Enum} type Definition.

© GIGABYTE™

Author:
shinanne.chen

Constructors

GbtDefinition

```
public  GbtDefinition()
```

Class GbtDefinition.GbtDevice

```
java.lang.Object
    |
    +--java.lang.Enum
        |
        +--Gbt.utility.API.GbtDefinition.GbtDevice
```

All Implemented Interfaces:
java.io.Serializable, java.lang.Comparable

< Fields > < Methods >

public static final class **GbtDefinition.GbtDevice**
extends java.lang.Enum

Device eunm for rackst and update CPLD command usage. The rmc and fanboard are used in {@link rackstDefinition}, and The bpb and mb are the devices for update CPLD which are used in {@link updateDefinition}.

Fields

bpb

```
public static final GbtDefinition.GbtDevice bpb
```

bpb device name used in {@link updateDefinition}

---

fanboard

```
public static final GbtDefinition.GbtDevice fanboard
```

fanboard device name used in {@link rackstDefinition#update}

---

mb

```
public static final GbtDefinition.GbtDevice mb
```

mother board device name used in {@link updateDefinition}

---

rmc

```
public static final GbtDefinition.GbtDevice rmc
```

rmc device name used in {@link rackstDefinition#update}

Methods

valueOf

```
public static GbtDefinition.GbtDevice valueOf(java.lang.String name)
```

---

values

```
public static Gbt.utility.API.GbtDefinition.GbtDevice[] values()
```

Class GbtDefinition.bmcutilDefinition

```
java.lang.Object
    |
    +--java.lang.Enum
        |
        +--Gbt.utility.API.GbtDefinition.bmcutilDefinition
```

All Implemented Interfaces:
java.io.Serializable, java.lang.Comparable

< Fields > < Methods >

---

public static final class **GbtDefinition.bmcutilDefinition**
extends java.lang.Enum

bmcutil command Definition of API arguments

Fields

SMTP

`public static final GbtDefinition.bmcutilDefinition` **`SMTP`**
Set BMC Web SMTP options by different setting method. detail see the following method

{@link bmcutilDefinition#setSMTPSenderInfo(String)} and {@link bmcutilDefinition#setSMTPEmail(int, boolean, String, String)} and {@link bmcutilDefinition#setSMTPserverIP(int, String)} and {@link bmcutilDefinition#setSMTPAuthority(boolean, String, String, String, String)}

SMTPSetAuthorityItem

`public static final java.lang.String` **`SMTPSetAuthorityItem`**

SMTPSetEmailItem

`public static final java.lang.String` **`SMTPSetEmailItem`**

SMTPSetSenderInfoItem

`public static final java.lang.String` **`SMTPSetSenderInfoItem`**

SMTPSetServerAddrItem

`public static final java.lang.String` **`SMTPSetServerAddrItem`**

SMTPitem

```
public static final java.lang.String SMTPitem
```

backup

```
public static final GbtDefinition.bmcutilDefinition backup
```

Download BMC configuration as backup file from BMC Web, please set command arguments by using {@link bmcutilDefinition#setDownloadBackupPath(String)} method.

backupitem

```
public static final java.lang.String backupitem
```

backupjs

```
public static final java.lang.String backupjs
```

prtsc

```
public static final GbtDefinition.bmcutilDefinition prtsc
```

Download upload last crash image from BMC Web, please set command arguments by using {@link bmcutilDefinition#setDownloadImgPath(String)} method.

prtscitem

```
public static final java.lang.String prtscitem
```

prtscjs

```
public static final java.lang.String prtscjs
```

reboot

```
public static final GbtDefinition.bmcutilDefinition reboot
```

reboot BMC

rebootBMCitem

```
public static final java.lang.String rebootBMCitem
```

---

rebootBMCjs

```
public static final java.lang.String rebootBMCjs
```

---

restore

```
public static final GbtDefinition.bmcutilDefinition restore
```

Restore BMC configuration to BMC Web, please set command arguments by using {@link bmcutilDefinition#setRestoreBackupFile(String)} method.

---

restoreitem

```
public static final java.lang.String restoreitem
```

---

restorejs

```
public static final java.lang.String restorejs
```

Methods

getArgs

```
public java.lang.String[] getArgs(GbtDefinition.bmcutilDefinition bmcutil)
```

Get Aarguments of vmcli command line which in vmcliDefinition.Mount or vmcliDefinition.Umount

Parameters:

bmcutil - : {@link bmcutilDefinition} defined options

---

getItem

```
public java.lang.String getItem()
```

Get function item

---

setDownloadBackupPath

`public GbtDefinition.bmcutilDefinition setDownloadBackupPath(java.lang.String path)`

Set Download BMC configuration file Path of backup BMC config

Parameters:

path - : Download config to given this path

---

setDownloadImgPath

`public GbtDefinition.bmcutilDefinition setDownloadImgPath(java.lang.String path)`

Set Download Image Path of Upload Last Crash Image

Parameters:

path - : Download Image to given this path

---

setRestoreBackupFile

`public GbtDefinition.bmcutilDefinition setRestoreBackupFile(java.lang.String file)`

Set restore BMC configuration file Path of retore BMC config

Parameters:

file - : Restore config by given this file

---

setSMTPAuthority

```
public GbtDefinition.bmcutilDefinition setSMTPAuthority(boolean Enable,
                                                        java.lang.String
Username,
                                                        java.lang.String
Password,
                                                        java.lang.String
STARTTLS_Mode,
                                                        java.lang.String
SASL_Mode)
```

Set BMC SMTP Authentication

Parameters:

Enable - : Enable (true) or Disable (false) the SMTP server
Username - :set SMTP server user account
Password - : set SMTP server password
STARTTLS_Mode - : There are 3 different STARTTLS Mode you can set as follow : auto, on, off
SASL_Mode - : There are 5 different SASL mode you can set as follow : auto, plain, login, htlm, md5

---

setSMTPEmail

```
public GbtDefinition.bmcutilDefinition setSMTPEmail(int AlertNum,
                                                    boolean Enable,
                                                    java.lang.String
ipAddress,
                                                    java.lang.String
Description)
```

Set BMC SMTP Destination Email Addresses

Parameters:

AlertNum - : Number of email destination as same as BMC Web.
Enable - : Enable (true) or disable (false) this email destination.
ipAddress - : destination IP address support IPv4 and IPv6.
Description - : email description (subject).

---

setSMTPSenderInfo

```
public GbtDefinition.bmcutilDefinition setSMTPSenderInfo(java.lang.String
Infomation)
```

Set BMC SMTP Sender Infomation

Parameters:

Infomation - : set SMTP sender info

---

setSMTPserverIP

```
public GbtDefinition.bmcutilDefinition setSMTPserverIP(int portNum,
                                                       java.lang.String
ipAddress)
```

Set BMC SMTP Server IP Address and port number

Parameters:

portNum - : SMTP server port.
ipAddress - : SMTP server IP address support IPv4 and IPv6.

---

valueOf

```
public static GbtDefinition.bmcutilDefinition valueOf(java.lang.String name)
```

---

values

```
public static Gbt.utility.API.GbtDefinition.bmcutilDefinition[] values()
```

Class GbtDefinition.loaddfDefinition

```
java.lang.Object
    |
    +--java.lang.Enum
        |
        +--Gbt.utility.API.GbtDefinition.loaddfDefinition
```

All Implemented Interfaces:
java.io.Serializable, java.lang.Comparable

< Fields > < Methods >

---

public static final class **GbtDefinition.loaddfDefinition**
extends java.lang.Enum

load default setting command argumets definition

Fields

BIOS

```
public static final GbtDefinition.loaddfDefinition BIOS
```

BIOS load default

BMC

```
public static final GbtDefinition.loaddfDefinition BMC
```

BMC load default

Cancel

```
public static final GbtDefinition.loaddfDefinition Cancel
```

Cancel

Methods

getItem

```
public java.lang.String getItem()
```

get {@link loaddfDefinition} defined item as {@link String}

valueOf

```
public static GbtDefinition.loaddfDefinition valueOf(java.lang.String name)
```

values

```
public static Gbt.utility.API.GbtDefinition.loaddfDefinition[] values()
```

Class GbtDefinition.rackexDefinition

```
java.lang.Object
    |
    +--java.lang.Enum
        |
        +--Gbt.utility.API.GbtDefinition.rackexDefinition
```

All Implemented Interfaces:

java.io.Serializable, java.lang.Comparable

< Fields > < Methods >

public static final class **GbtDefinition.rackexDefinition**
extends java.lang.Enum

GbtUtility rackex command definition for API usage. rackex command support RMC 2.0. There are three sub oprions "update", "dump" and "unlock" you can used.

Fields

dump

`public static final GbtDefinition.rackexDefinition` **`dump`**

Dump racklution-EX whole rack log.

---

unlock

`public static final GbtDefinition.rackexDefinition` **`unlock`**

Unlock abnormal rmc.

---

update

`public static final GbtDefinition.rackexDefinition` **`update`**

Update racklution-EX firmware(rmc and control board). corresponding arguments set method : {@link rackexDefinition#setUpdateArgs}

Methods

getArgs

`public java.lang.String[]` **`getArgs`**`(GbtDefinition.rackexDefinition action)`

get arguments of action

Parameters:

action - : {@link rackexDefinition} defined options.

---

getItem

`public java.lang.String` **`getItem`**`()`

get {@link rackexDefinition} defined item as {@link String}

---

setUpdateArgs

```
public GbtDefinition.rackexDefinition setUpdateArgs(java.lang.String Firmware,
                                                    boolean auto,
                                                    java.lang.String
MACList_path)
```

Set RMC 2.0 update arguments methods

Parameters:

Firmware - : RMC Firmware package, image package only support .zip file.
auto - : (Optoinal)Set this argument to skip yes no question.(Yes to all).
MACList_path - : (Optoinal)For checking target mac list, only all mac include scan scope will keep update.

---

valueOf

```
public static GbtDefinition.rackexDefinition valueOf(java.lang.String name)
```

---

values

```
public static Gbt.utility.API.GbtDefinition.rackexDefinition[] values()
```

Class GbtDefinition.rackstDefinition

```
java.lang.Object
    |
    +--java.lang.Enum
        |
        +--Gbt.utility.API.GbtDefinition.rackstDefinition
```

All Implemented Interfaces:
java.io.Serializable, java.lang.Comparable

< Fields > < Methods >

---

public static final class **GbtDefinition.rackstDefinition**
extends java.lang.Enum

GbtUtility rackst command definition for API usage. rackst command support RMC 1.0. There are two sub oprions "update" and "manage" you can used.

Fields

manage

`public static final GbtDefinition.rackstDefinition` **`manage`**

The "manage" option can send rmc command to rmc. And you can called method {@link rackstDefinition#setRMCcommand} to set required arguments

---

update

`public static final GbtDefinition.rackstDefinition` **`update`**

The "update" option can update firmware of two devices "rmc" and "fanboard" which you can called in {@link GbtDevice}. And you can called method {@link rackstDefinition#setDevAndFW} to set required arguments

Methods

getArgs

`public java.lang.String[]` **`getArgs`**`(GbtDefinition.rackstDefinition rackstcmd)`

get arguments of rackst

Parameters:

rackstcmd - : {@link rackstDefinition} defined options.

---

getItem

`public java.lang.String` **`getItem`**`()`

get {@link rackstDefinition} defined item as {@link String}

---

setDevAndFW

`public GbtDefinition.rackstDefinition` **`setDevAndFW`**`(GbtDefinition.GbtDevice dev, java.lang.String firmwarename)`

rackst update command set device and update firmware file name for API

Parameters:

dev - : The target device for update option. device name defined in {@link GbtDevice}
firmwarename - : update firmware name

---

setRMCcommand

`public GbtDefinition.rackstDefinition setRMCcommand(java.lang.String cmd)`

rackst set rmc1.0 command for API

Parameters:

cmd - : rmc 1.0 command

---

valueOf

`public static GbtDefinition.rackstDefinition valueOf(java.lang.String name)`

---

values

`public static Gbt.utility.API.GbtDefinition.rackstDefinition[] values()`

Class GbtDefinition.updateDefinition

```
java.lang.Object
    |
    +--java.lang.Enum
        |
        +--Gbt.utility.API.GbtDefinition.updateDefinition
```

All Implemented Interfaces:

java.io.Serializable, java.lang.Comparable

< Fields > < Methods >

---

public static final class **GbtDefinition.updateDefinition**
extends java.lang.Enum

GbtUtility node devices firmware update command argumets definition

Fields

updBIOS

`public static final GbtDefinition.updateDefinition updBIOS`

update node BIOS firmware, please using {@link updateDefinition#setBIOSupdateArgs} to set your update argumets

---

updBMC

`public static final GbtDefinition.updateDefinition` **`updBMC`**

update node BMC firmware, please using {@link updateDefinition#setBMCupdateArgs} to set your update argumets

---

updCPLD

`public static final GbtDefinition.updateDefinition` **`updCPLD`**

update node CPLD firmware, please using {@link updateDefinition#setCPLDupdateArgs(GbtDevice, String)} or {@link updateDefinition#setCPLDupdateArgs(GbtDevice, String, String, String)} to set your update argumets

---

updPSU

`public static final GbtDefinition.updateDefinition` **`updPSU`**

update node PSU firmware, please using {@link updateDefinition#setPSUupdateArgs(String)} or {@link updateDefinition#setPSUupdateArgs(String, String, String)} to set your update argumets

Methods

getArgs

`public java.lang.String[]` **`getArgs`**`(GbtDefinition.updateDefinition dev)`

get arguments of rackst

Parameters:

dev - : {@link updateDefinition} defined options of update type.

---

getItem

`public java.lang.String` **`getItem`**`()`

get {@link updateDefinition} defined item as {@link String}

---

setBIOSupdateArgs

```
public GbtDefinition.updateDefinition setBIOSupdateArgs(java.lang.String[]
firmwares,
                                                        int autoMode,
                                                        java.lang.String
tftpPath,
                                                        java.lang.String
tftpIP,
                                                        boolean Force)
```

Set BIOS FW update arguments with given TFTP server IP and path which mode you want to use.

Notice that the TFTP server and target node which you wanted to update firmware must at the same DNS.

If you want to auto create TFTP server during the update process, then set tftpPath and tftpIP as null.

And make sure the port 69 is free.

Parameters:

firmwares - : BIOS update firmwares only support .zip or .tgz file.
autoMode - : 0- full manual mode.(default), utility will ask retry or stop if error happen.

1- Auto exit, utility will stop if error happen.

2- Auto retry, utility will auto retry if error happen.

tftpPath - : TFTP server root path
tftpIP - : TFTP server host IP
Force - : If this terms true, update process will skip compare firmware version and node fru, it will

force update the same version of firmware.

setBMCupdateArgs

```
public GbtDefinition.updateDefinition setBMCupdateArgs(java.lang.String[]
firmwares,
                                                       int autoMode,
                                                       java.lang.String
tftpPath,
                                                       java.lang.String
tftpIP,
                                                       boolean Force)
```

Set BMC FW update arguments with given TFTP server IP and path which mode you want to use.

Notice that the TFTP server and target node which you wanted to update firmware must at the same DNS.

If you want to auto create TFTP server during the update process, then set tftpPath and tftpIP as null.

And make sure the port 69 is free.

Parameters:

firmwares - : BMC update firmwares only support .zip or .tgz file.
autoMode - : 0- full manual mode.(default), utility will ask retry or stop if error happen.

1- Auto exit, utility will stop if error happen.

2- Auto retry, utility will auto retry if error happen.

tftpPath - : TFTP server root path
tftpIP - : TFTP server host IP
Force - : If this terms true, update process will skip compare firmware version and node fru, it will

force update the same version of firmware.

---

setCPLDupdateArgs

```
public GbtDefinition.updateDefinition
setCPLDupdateArgs(GbtDefinition.GbtDevice device,
                                                       java.lang.String
CPLDfirmware)
```

Set CPLD update arguments with auto initialize TFTP server setting and open TFTP for update.

This arguments set method will help you auto create TFTP server during the update process.

To make sure the port 69 is free and notice that this computer and target node which

you wanted to update firmware must at the same DNS.

Parameters:

device - : update devices difined in {@link GbtDevice}
CPLDfirmware - : CPLD update firmware name

---

setCPLDupdateArgs

```
public GbtDefinition.updateDefinition
setCPLDupdateArgs(GbtDefinition.GbtDevice device,
                  java.lang.String CPLDfirmware,
                  java.lang.String TFTPseverIP,
                  java.lang.String TFTPpath)
```

Set CPLD FW update arguments with given TFTP server IP and path which you want to use.

Notice that the TFTP server and target node which you wanted to update firmware must at the same DNS.

Parameters:

device - : update devices difined in {@link GbtDevice}
CPLDfirmware - : CPLD update firmware name
TFTPseverIP - : TFTP server host IP
TFTPpath - : TFTP server root path

---

setPSUupdateArgs

```
public GbtDefinition.updateDefinition setPSUupdateArgs(java.lang.String PSUfirmware)
```

Set PSU update arguments with auto initialize TFTP server setting and open TFTP for update.

This arguments set method will help you auto create TFTP server during the update process.

To make sure the port 69 is free and notice that this computer and target node which

you wanted to update firmware must at the same DNS.

Parameters:

PSUfirmware - : PSU firmware name

---

setPSUupdateArgs

```
public GbtDefinition.updateDefinition setPSUupdateArgs(java.lang.String PSUfirmware,
                  java.lang.String TFTPseverIP,
                  java.lang.String TFTPpath)
```

Set PSU FW update arguments with given TFTP server IP and path which you want to use.

Notice that the TFTP server and target node which you wanted to update firmware must at the same DNS.

Parameters:

PSUfirmware - : PSU firmware name
TFTPseverIP - : TFTP server host IP
TFTPpath - : TFTP server root path

valueOf

```
public static GbtDefinition.updateDefinition valueOf(java.lang.String name)
```

values

```
public static Gbt.utility.API.GbtDefinition.updateDefinition[] values()
```

Class GbtDefinition.userappDefinition

```
java.lang.Object
    |
    +--java.lang.Enum
        |
        +--Gbt.utility.API.GbtDefinition.userappDefinition
```

All Implemented Interfaces:
java.io.Serializable, java.lang.Comparable

< Fields > < Methods >

public static final class **GbtDefinition.userappDefinition**
extends java.lang.Enum

GbtUtility node BMC User Account command arguments definition

Fields

List

```
public static final GbtDefinition.userappDefinition List
```
List all BMC User Accounts and their detail information

Set

```
public static final GbtDefinition.userappDefinition Set
```
Set username, password and IPMI {@link PrivilegeLevel} at specific userid, please using {@link userappDefinition#setAccount(String, String, String, PrivilegeLevel)} method to set userapp command arguments

Methods

getArgs

```
public java.lang.String[] getArgs(GbtDefinition.userappDefinition subcmd)
```

get arguments of rackst

Parameters:

subcmd - : {@link userappDefinition} defined options.

---

getItem

```
public java.lang.String getItem()
```

get {@link userappDefinition} defined item as {@link String}

---

setAccount

```
public GbtDefinition.userappDefinition setAccount(java.lang.String username,
                                                  java.lang.String password,
                                                  java.lang.String userid,
com.veraxsystems.vxipmi.coding.commands.PrivilegeLevel priv)
```

Set user account arguments

Parameters:

username - : user name
password - : user password
userid - : BMC user id from 3 to 16
priv - : IPMI Privilege Level defined in {@link PrivilegeLevel}

---

valueOf

```
public static GbtDefinition.userappDefinition valueOf(java.lang.String name)
```

---

values

```
public static Gbt.utility.API.GbtDefinition.userappDefinition[] values()
```

Class GbtDefinition.vmcliDefinition

```
java.lang.Object
    |
    +--java.lang.Enum
        |
        +--Gbt.utility.API.GbtDefinition.vmcliDefinition
```

All Implemented Interfaces:
java.io.Serializable, java.lang.Comparable

< Fields > < Methods >

---

public static final class **GbtDefinition.vmcliDefinition**
extends java.lang.Enum

GbtUtility vmcli command Definition

Fields

Mount

`public static final GbtDefinition.vmcliDefinition` **`Mount`**
Mount virtual media to remote server by BMC, please using {@link vmcliDefinition#setVertualMediaDevices(String, String, String)} method to set virtual media which you want to mount.

Umount

`public static final GbtDefinition.vmcliDefinition` **`Umount`**
Umount virtual media from remote server by BMC

Methods

getArgs

`public java.lang.String[]` **`getArgs`**`(GbtDefinition.vmcliDefinition vmcli)`

Get Aarguments of vmcli command line which in vmcliDefinition.Mount or vmcliDefinition.Umount

Parameters:

vmcli - : {@link vmcliDefinition} defined options

---

getItem

```
public java.lang.String getItem()
```

get {@link vmcliDefinition} defined item as {@link String}

---

setVertualMediaDevices

```
public GbtDefinition.vmcliDefinition setVertualMediaDevices(java.lang.String
isoName,
                                                            java.lang.String
floppyImg,
                                                            java.lang.String
removableDev)
```

Set ISO, floppy, removable devices or image files of vmcli Mount option. If the device ex : floppyImg you won't set any device (image file) , please set "" for this argument

Parameters:

isoName - : ISO image file or CD/DVD ROM device name
floppyImg - : Floppy image file or Floppy device name
removableDev - : removable device name ex : USB disk

---

valueOf

```
public static GbtDefinition.vmcliDefinition valueOf(java.lang.String name)
```

---

values

```
public static Gbt.utility.API.GbtDefinition.vmcliDefinition[] values()
```

Class GbtUtilityAPI

```
java.lang.Object
    |
    +--Gbt.utility.API.GbtUtilityAPI
```

< Constructors > < Methods >

---

public class **GbtUtilityAPI**
extends java.lang.Object

GbtUtility Command API

© GIGABYTE™

Constructors

GbtUtilityAPI

```
public  GbtUtilityAPI()
```

Load GbtUtility initial properties

For example:

```
GbtUtilityAPI gbt = new GbtUtilityAPI();
gbt.Scan();
```

Methods

Chassis

```
public void Chassis(java.lang.String action)
```

chassis by using Nodelist,

For example:

- Chassis("on");

Parameters:

action - the action of chassis

Chassis

```
public void Chassis(java.lang.String bmcUserName,
                    java.lang.String bmcUserPasswd,
                    java.lang.String action,
                    java.lang.String ip)
```

chassis for single node,

For example:

- Chassis("on","10.2.2.10");
- Chassis("off","10.2.2.10");

Parameters:

action - the action of chassis
ip - the ip of node

Chassis

```
public void Chassis(java.lang.String bmcUserName,
                    java.lang.String bmcUserPasswd,
                    java.lang.String action,
                    java.lang.String startip,
                    java.lang.String endip,
                    java.lang.String rackType)
```

chassis by scan ip range,

For example:

- Chassis("on","10.2.2.10","10.2.2.98","Standard");
- Chassis("off","10.2.2.10","10.2.2.98","OCP");
- Chassis("cycle","10.2.2.10","10.2.2.98","OR25");
- Chassis("reset","10.2.2.10","10.2.2.98","CMC");

Parameters:

action - the action of chassis
startip - the ip of range start
endip - the ip of range end
rackType - Type of Rack

JSONParser

```
public void JSONParser(java.lang.String location)
```

Using JSON file to excute the GBTUtility command

For example:

- JSONParser("./jsonTest1.json")

Parameters:

location - the path of JSON file

LoadNodeList

```
public boolean LoadNodeList()
```

Check Nodelist exist or not

Returns:

true if Nodelist exist, false otherwise.

LocalSaveDefault

```
public void LocalSaveDefault()
```

Local save Default. The following usage hints of this method will guide you how to use.

For Example:

- GbtUtilityAPI gbt = new GbtUtilityAPI();
- gbt.LocalSaveDefault();

Scan

```
public void Scan(java.lang.String startip,
                 java.lang.String endip,
                 java.lang.String rackType)
```

Scan Node IP from a range

For example:

- Scan("10.2.2.10","10.2.2.98","Standard")

Parameters:

startip - the ip of range start
endip - the ip of range end
rackType - Type of Rack

---

bmcutil

`public void` **`bmcutil`**`(GbtDefinition.bmcutilDefinition action)`

BMC Web Utility API

usage hints of this method will guide you how to use.

For Example :

reboot BMC :

- GbtUtilityAPI gbt = new GbtUtilityAPI();
- gbt.bmcutil(bmcutilDefinition.reboot);

Upload Last Crash Image :

- GbtUtilityAPI gbt = new GbtUtilityAPI();
- gbt.bmcutil(bmcutilDefinition.prtsc.setDownloadImgPath("/home/pictures"));

SMTP :

- GbtUtilityAPI gbt = new GbtUtilityAPI();
- gbt.bmcutil(bmcutilDefinition.SMTP.setSMTPSenderInfo("yuor information."));
- gbt.bmcutil(bmcutilDefinition.SMTP.setSMTPEmail(1, true, "10.2.2.2", "your description"));
- gbt.bmcutil(bmcutilDefinition.SMTP.setSMTPserverIP(2050, "10.2.1.1"));
- gbt.bmcutil(bmcutilDefinition.SMTP.setSMTPAuthority(true, "GIGABYTE", "password", "auto", "auto"));

Parameters:

set - arguments of bmcutil the first position String is sub command just like command line utility.

bmcutil

```
public void bmcutil(java.lang.String bmcUserName,
                    java.lang.String bmcUserPasswd,
                    GbtDefinition.bmcutilDefinition action,
                    java.lang.String ip)
```

BMC Web Utility API

usage hints of this method will guide you how to use.

For Example :

reboot BMC :

- GbtUtilityAPI gbt = new GbtUtilityAPI();
- gbt.bmcutil("admin","password",bmcutilDefinition.reboot,"10.1.1.25");

Upload Last Crash Image :

- GbtUtilityAPI gbt = new GbtUtilityAPI();
- gbt.bmcutil("admin","password",bmcutilDefinition.prtsc.setDownloadImgPath("/home/pictures"),"10.1.1.25

SMTP :

- GbtUtilityAPI gbt = new GbtUtilityAPI();
- gbt.bmcutil("admin","password",bmcutilDefinition.SMTP.setSMTPSenderInfo("yuor information."),"10.1.1.25");
- gbt.bmcutil("admin","password",bmcutilDefinition.SMTP.setSMTPEmail(1, true, "10.2.2.2", "your description"),"10.1.1.25");
- gbt.bmcutil("admin","password",bmcutilDefinition.SMTP.setSMTPserverIP(2050, "10.2.1.1"),"10.1.1.25");
- gbt.bmcutil("admin","password",bmcutilDefinition.SMTP.setSMTPAuthority(true, "GIGABYTE", "password", "auto", "auto"),"10.1.1.25");

Parameters:

set - arguments of bmcutil the first position String is sub command just like command line utility.

---

bmcutil

```
public void bmcutil(java.lang.String bmcUserName,
                    java.lang.String bmcUserPasswd,
                    GbtDefinition.bmcutilDefinition action,
                    java.lang.String startip,
                    java.lang.String endip,
                    java.lang.String rackType)
```

BMC Web Utility API

usage hints of this method will guide you how to use.

For Example :

reboot BMC :

- GbtUtilityAPI gbt = new GbtUtilityAPI();
- gbt.bmcutil("admin","password",
  bmcutilDefinition.reboot,"10.1.1.25","10.1.1.50","Standard");

Upload Last Crash Image :

- GbtUtilityAPI gbt = new GbtUtilityAPI();
- gbt.bmcutil("admin","password",

bmcutilDefinition.prtsc.setDownloadImgPath("/home/pictures"),"10.1.1.25","10.1.1.50","Standard");

SMTP :

- GbtUtilityAPI gbt = new GbtUtilityAPI();
- gbt.bmcutil("admin","password",bmcutilDefinition.SMTP.setSMTPSenderInfo("yuor information."),"10.1.1.25","10.1.1.50","Standard");
- gbt.bmcutil("admin","password",bmcutilDefinition.SMTP.setSMTPEmail(1, true, "10.2.2.2", "your description"),"10.1.1.25","10.1.1.50","Standard");
- gbt.bmcutil("admin","password",bmcutilDefinition.SMTP.setSMTPserverIP(2050, "10.2.1.1"),"10.1.1.25","10.1.1.50","Standard");
- gbt.bmcutil("admin","password",bmcutilDefinition.SMTP.setSMTPAuthority(true, "GIGABYTE", "password", "auto", "auto"),"10.1.1.25","10.1.1.50","Standard");

Parameters:

set - arguments of bmcutil the first position String is sub command just like command line utility.

bmcver

```
public void bmcver()
```

Send IPMI command to BMC and get FW revision of BMC with several IPs, which are contained on NodeList.log file. If NodeList.log file does not exist or you wanted to change IP range. Please using Scan method create a new NodeList.log file before using this method. Or you can just using another argument method. {@link GbtUtilityAPI#bmcver(String, String, String)}. The execute message will be written into each IP's log file. Ex : 10.2.7.13_BMCVersion.log. The following usage hints of this method will guide you how to use.

For Example:

- GbtUtilityAPI gbt = new GbtUtilityAPI();
- gbt.bmcver();

---

bmcver

```
public void bmcver(java.lang.String bmcUserName,
                   java.lang.String bmcUserPasswd,
                   java.lang.String ip)
```

Send IPMI command to BMC and get FW revision of BMC with single IP. The execute message will be written into each IP's log file. Ex : 10.2.7.13_BMCVersion.log. The following usage hints of this method will guide you how to use.

For Example:

- GbtUtilityAPI gbt = new GbtUtilityAPI();
- gbt.bmcver("10.2.2.1");

Parameters:

ip - single BMC IP

---

bmcver

```
public void bmcver(java.lang.String bmcUserName,
                   java.lang.String bmcUserPasswd,
                   java.lang.String startip,
                   java.lang.String endip,
                   java.lang.String rackType)
```

Send IPMI command to BMC and get FW revision of BMC with several IPs, which are contained on NodeList.log file. The scan option will scan IPs between startip and endip. After scan these IP will be written into NodeList.log. The execute message will be written into each IP's log file. Ex : 10.2.7.13_BMCVersion.log. The following usage hints of this method will guide you how to use.

For Example:

- GbtUtilityAPI gbt = new GbtUtilityAPI();
- gbt.bmcver("10.2.2.1", "10.2.2.100" , "Standard");

Parameters:

startip - start BMC IP for scan option
endip - end BMC IP for scan option
rackType - rack type of node server default using "Standard"

---

boot

```
public void boot(java.lang.String action)
```

select boot device by using nodelist

For example:

- boot("none");
- boot("pxe");
- boot("cdrom");

Parameters:

action - the action of limit

---

boot

```
public void boot(java.lang.String bmcUserName,
                 java.lang.String bmcUserPasswd,
                 java.lang.String action,
                 java.lang.String ip)
```

select boot device for single node

For example:

- boot("none","10.2.2.10");
- boot("pxe","10.2.2.10");
- boot("cdrom","10.2.2.10");

Parameters:

action - the action of limit
ip - the ip of node

---

boot

```
public void boot(java.lang.String bmcUserName,
                 java.lang.String bmcUserPasswd,
                 java.lang.String action,
                 java.lang.String startip,
                 java.lang.String endip,
                 java.lang.String rackType)
```

select boot device by scan ip range

For example:

- boot("none","10.2.2.10","10.2.2.98","Standard");
- boot("pxe","10.2.2.10","10.2.2.98","OCP");
- boot("cdrom","10.2.2.10","10.2.2.98","OCP");

Parameters:

action - the action of limit
startip - the ip of range start
endip - the ip of range end
rackType - Type of Rack

---

fru

```
public void fru()
```

Show fru data by using Nodelist

For example:

- fru("10.2.2.10");

---

fru

```
public void fru(java.lang.String bmcUserName,
                java.lang.String bmcUserPasswd,
                java.lang.String ip)
```

Show fru data for single node

For example:

- fru("10.2.2.10");

Parameters:

startip - the ip of node

---

fru

```
public void fru(java.lang.String bmcUserName,
                java.lang.String bmcUserPasswd,
                java.lang.String startip,
                java.lang.String endip,
                java.lang.String rackType)
```

Show fru data by scan ip range

For example:

- fru("10.2.2.10","10.2.2.98","Standard");
- fru("10.2.2.10","10.2.2.98","OCP");

Parameters:

startip - the ip of range start
endip - the ip of range end
rackType - Type of Rack

---

hwinfo

```
public void hwinfo()
```

Show hwinfo info by using Nodelist

For example:

- hwinfo();

---

hwinfo

```
public void hwinfo(java.lang.String bmcUserName,
                   java.lang.String bmcUserPasswd,
                   java.lang.String ip)
```

Show hwinfo info for single node

For example:

- hwinfo("10.2.2.10");
- hwinfo("10.2.2.10");

Parameters:

ip - the ip of node

---

hwinfo

```
public void hwinfo(java.lang.String bmcUserName,
                   java.lang.String bmcUserPasswd,
                   java.lang.String startip,
                   java.lang.String endip,
                   java.lang.String rackType)
```

Show hwinfo info by scan ip range

For example:

- hwinfo("10.2.2.10","10.2.2.98","Standard");
- hwinfo("10.2.2.10","10.2.2.98","OCP");

Parameters:

startip - the ip of range start
endip - the ip of range end
rackType - Type of Rack

---

identify

```
public void identify(java.lang.String interval)
```

Control node panel identify light with several BMC IPs, which are contained on NodeList.log file. If NodeList.log file does not exist or you wanted to change IP range. Please using Scan method create a new NodeList.log file before using this method. Or you can just using another argument method. {@link GbtUtilityAPI#identify(String, String, String, String)}. The following usage hints of this method will guide you how to use.

For Example:

- GbtUtilityAPI gbt = new GbtUtilityAPI();
- gbt.identify("200"); //Turn on light with this interval(Unit: second)
- gbt.identify("0"); //Turn off light.

Parameters:

interval - String type. Turn on light with this interval 1~255 (Unit: second), 0 = turn off light

---

identify

```
public void identify(java.lang.String bmcUserName,
                     java.lang.String bmcUserPasswd,
                     java.lang.String interval,
                     java.lang.String ip)
```

Control node panel identify light with single BMC IP. The following usage hints of this method will guide you how to use.

For Example:

- GbtUtilityAPI gbt = new GbtUtilityAPI();
- gbt.bmcver("200","10.2.2.1"); //Turn on light with this interval(Unit: second)
- gbt.identify("0","10.2.2.1"); //Turn off light.

Parameters:

ip - single BMC IP

---

identify

```
public void identify(java.lang.String bmcUserName,
                     java.lang.String bmcUserPasswd,
                     java.lang.String interval,
                     java.lang.String startip,
                     java.lang.String endip,
                     java.lang.String rackType)
```

Control node panel identify light with several BMC IPs, which are contained on NodeList.log file. The scan option will scan IPs between startip and endip. After scan these IP will be written into NodeList.log. The following usage hints of this method will guide you how to use.

For Example:

- GbtUtilityAPI gbt = new GbtUtilityAPI();
- gbt.identify(200, "10.2.2.1", "10.2.2.100" , "Standard"); //Turn on light with this interval(Unit: second)
- gbt.identify(0, "10.2.2.1", "10.2.2.100" , "Standard"); //Turn off light.

Parameters:

interval - String type. Turn on light with this interval 1~255 (Unit: second), 0 = turn off light
startip - start BMC IP for scan option
endip - end BMC IP for scan option
rackType - rack type of node server default using "Standard"

---

ikvm

```
public void ikvm(java.lang.String bmcUserName,
                 java.lang.String bmcUserPasswd,
                 java.lang.String ip)
```

Launch screen view with single BMC IP. The following usage hints of this method will guide you how to use.

For Example:

- GbtUtilityAPI gbt = new GbtUtilityAPI();
- gbt.ikvm("10.2.2.1");

Parameters:

ip - single BMC IP

---

ipsrc

```
public void ipsrc(java.lang.String action,
                  java.lang.String option)
```

Show ip source by using Nodelist

For example:

- ipsrc("get","DHCP");
- ipsrc("set","Static");

Parameters:

action - input bytes
option - DHCP|Static(set only)

---

ipsrc

```
public void ipsrc(java.lang.String bmcUserName,
                  java.lang.String bmcUserPasswd,
                  java.lang.String action,
                  java.lang.String ip,
                  java.lang.String option)
```

Show ip source for single node

For example:

- ipsrc("get","10.2.2.10","DHCP");
- ipsrc("set","10.2.2.10","Static");

Parameters:

action - input bytes
ip - the ip of node
option - DHCP|Static(set only)

---

ipsrc

```
public void ipsrc(java.lang.String bmcUserName,
                  java.lang.String bmcUserPasswd,
                  java.lang.String action,
                  java.lang.String startip,
                  java.lang.String endip,
                  java.lang.String rackType,
                  java.lang.String option)
```

Show ip source by scan ip range

For example:

- ipsrc("get","10.2.2.10","10.2.2.98","Standard","DHCP");
- ipsrc("set","10.2.2.10","10.2.2.98","Static");

Parameters:

action - input bytes
startip - the ip of range start
endip - the ip of range end
rackType - Type of Rack
option - DHCP|Static(set only)

---

ipv6

```
public void ipv6(java.lang.String action)
```

target node ipv6 config by using Nodelist

For example:

- ipv6("enable");
- ipv6("disable");

Parameters:

enable/disable - target node ipv6 config.

---

ipv6

```
public void ipv6(java.lang.String bmcUserName,
                 java.lang.String bmcUserPasswd,
                 java.lang.String action,
                 java.lang.String ip)
```

target node ipv6 config for single node

For example:

- ipv6("enable","10.2.2.10");
- ipv6("disable","10.2.2.10");

Parameters:

enable/disable - target node ipv6 config.
ip - the ip of node

---

ipv6

```
public void ipv6(java.lang.String bmcUserName,
                 java.lang.String bmcUserPasswd,
                 java.lang.String action,
                 java.lang.String startip,
                 java.lang.String endip,
                 java.lang.String rackType)
```

target node ipv6 config by scan ip range

For example:

- ipv6("enable","10.2.2.10","10.2.2.98","Standard");
- ipv6("disable","10.2.2.10","10.2.2.98","OCP");

Parameters:

action - enable/disable target node ipv6 config.
startip - the ip of range start
endip - the ip of range end
rackType - Type of Rack

---

limit

```
public void limit(java.lang.String action,
                  int limitValue)
```

set/deactivate target node power limit by using Nodelist

For example:

- limit("set",220);
- limit("deactivate",0);

Parameters:

action - the action of limit
limitValue - power limit value. If action is deactivate,then put 0 for this parameter.

---

limit

```
public void limit(java.lang.String bmcUserName,
                  java.lang.String bmcUserPasswd,
                  java.lang.String action,
                  java.lang.String ip,
                  int limitValue)
```

set/deactivate target node power limit for single node

For example:

- limit("set","10.2.2.10",220);
- limit("deactivate","10.2.2.10",0);

Parameters:

action - the action of limit
ip - the ip of node
limitValue - power limit value. If action is deactivate,then put 0 for this parameter.

---

limit

```
public void limit(java.lang.String bmcUserName,
                  java.lang.String bmcUserPasswd,
                  java.lang.String action,
                  java.lang.String startip,
                  java.lang.String endip,
                  java.lang.String rackType,
                  int limitValue)
```

set/deactivate target node power limit for ip range

For example:

- limit("set","10.2.2.10","10.2.2.98","Standard",220);
- limit("deactivate","10.2.2.10","10.2.2.98","OCP",0);

Parameters:

action - the action of limit
startip - the ip of range start
endip - the ip of range end
rackType - Type of Rack
limitValue - power limit value. If action is deactivate,then put 0 for this parameter.

---

loaddf

```
public void loaddf(GbtDefinition.loaddfDefinition dev)
```

Load default to nodes (load to bmc/bios/cnacel) with several BMC IPs, which are contained on NodeList.log file. If NodeList.log file does not exist or you wanted to change IP range. Please using Scan method create a new NodeList.log file before using this method. Or you can just using another argument method. {@link GbtUtilityAPI#loaddf(loaddfDefinition, String, String, String)}. The following usage hints of this method will guide you how to use.

For Example:

- GbtUtilityAPI gbt = new GbtUtilityAPI();
- gbt.loaddf(loaddfDefinition.BIOS);
- gbt.loaddf(loaddfDefinition.BMC);
- gbt.loaddf(loaddfDefinition.Cancel);

Parameters:

dev - loaddfDefinition type of load default option (tag has BIOS BMC and Cancel)

---

loaddf

```
public void loaddf(java.lang.String bmcUserName,
                   java.lang.String bmcUserPasswd,
                   GbtDefinition.loaddfDefinition dev,
                   java.lang.String ip)
```

Load default to node (load to bmc/bios/cnacel) with single BMC IP. The following usage hints of this method will guide you how to use.

For Example:

- GbtUtilityAPI gbt = new GbtUtilityAPI();
- gbt.loaddf(loaddfDefinition.BIOS, "10.2.2.1");
- gbt.loaddf(loaddfDefinition.BMC, "10.2.2.1");
- gbt.loaddf(loaddfDefinition.Cancel, "10.2.2.1");

Parameters:

dev - loaddfDefinition type of load default option (tag has BIOS BMC and Cancel)
ip - single BMC IP

---

loaddf

```
public void loaddf(java.lang.String bmcUserName,
                   java.lang.String bmcUserPasswd,
                   GbtDefinition.loaddfDefinition dev,
                   java.lang.String startip,
                   java.lang.String endip,
                   java.lang.String rackType)
```

Load default to nodes (load to bmc/bios/cnacel) with several BMC IPs, which are contained on NodeList.log file. The scan option will scan IPs between startip and endip. After scan these IP will be written into NodeList.log. The following usage hints of this method will guide you how to use.

For Example:

- GbtUtilityAPI gbt = new GbtUtilityAPI();
- gbt.loaddf(loaddfDefinition.BIOS, "10.2.2.1", "10.2.2.1", "10.2.2.100" , "Standard");
- gbt.loaddf(loaddfDefinition.BMC, "10.2.2.1", "10.2.2.1", "10.2.2.100" , "Standard");
- gbt.loaddf(loaddfDefinition.Cancel, "10.2.2.1", "10.2.2.1", "10.2.2.100" , "Standard");

Parameters:

dev - loaddfDefinition type of load default option (tag has BIOS BMC and Cancel)
startip - start BMC IP for scan option
endip - end BMC IP for scan option
rackType - rack type of node server default using "Standard"

---

logochange

```
public void logochange(java.lang.String bmcUserName,
                       java.lang.String bmcUserPasswd,
                       java.lang.String[] imagefiles,
                       java.lang.String ip)
```

Change LOGO with single BMC IP. The following usage hints of this method will guide you how to use.

For Example:

- GbtUtilityAPI gbt = new GbtUtilityAPI();
- String[] tmp = {LOGO1.png, LOGO2.png};
- gbt.logochange("tmp", "10.2.2.1");

Parameters:

imagefiles - LOGO image names
ip - single BMC IP

---

logochange

```
public void logochange(java.lang.String bmcUserName,
                       java.lang.String bmcUserPasswd,
                       java.lang.String[] imagefiles,
                       java.lang.String startip,
                       java.lang.String endip,
                       java.lang.String rackType)
```

Change LOGO with several BMC IPs, which are contained on NodeList.log file. The scan option will scan IPs between startip and endip. After scan these IP will be written into NodeList.log. The following usage hints of this method will guide you how to use.

For Example:

- GbtUtilityAPI gbt = new GbtUtilityAPI();
- String[] tmp = {LOGO1.png, LOGO2.png};
- gbt.logochange("tmp", "10.2.2.1", "10.2.2.1", "10.2.2.100" , "Standard");

Parameters:

imagefiles - LOGO image names
startip - start BMC IP for scan option
endip - end BMC IP for scan option
rackType - rack type of node server default using "Standard"

---

logochange

```
public void logochange(java.lang.String[] imagefiles)
```

Change LOGO with several BMC IPs, which are contained on NodeList.log file. If NodeList.log file does not exist or you wanted to change IP range. Please using Scan method create a new NodeList.log file before using this method. Or you can just using another argument method. {@link GbtUtilityAPI#logochange(String[], String, String, String)}. The following usage hints of this method will guide you how to use.

For Example:

- GbtUtilityAPI gbt = new GbtUtilityAPI();
- String[] tmp = {LOGO1.png, LOGO2.png};
- gbt.logochange("tmp");

Parameters:

imagefiles - LOGO image names

---

mesensor

```
public void mesensor()
```

mesensor by using Nodelist

For example:

- mesensor();

---

mesensor

```
public void mesensor(java.lang.String bmcUserName,
                     java.lang.String bmcUserPasswd,
                     java.lang.String ip)
```

mesensor for single node

For example:

- mesensor("10.2.2.10");
- mesensor("10.2.2.10");

Parameters:

ip - the ip of node

---

mesensor

```
public void mesensor(java.lang.String bmcUserName,
                     java.lang.String bmcUserPasswd,
                     java.lang.String startip,
                     java.lang.String endip,
                     java.lang.String rackType)
```

mesensor by scan ip range

For example:

- mesensor("10.2.2.10","10.2.2.98","Standard");
- mesensor("10.2.2.10","10.2.2.98","OCP");

Parameters:

startip - the ip of range start
endip - the ip of range end
rackType - Type of Rack

---

msr

```
public void msr()
```

Show all register status diagnosis by using Nodelist

For example:

- msr();

---

msr

```
public void msr(java.lang.String bmcUserName,
                java.lang.String bmcUserPasswd,
                java.lang.String ip)
```

Show all register status diagnosis for single node

For example:

- msr("10.2.2.10");
- msr("10.2.2.10");

Parameters:

ip - the ip of node

---

msr

```
public void msr(java.lang.String bmcUserName,
                java.lang.String bmcUserPasswd,
                java.lang.String startip,
                java.lang.String endip,
                java.lang.String rackType)
```

Show all register status diagnosis by scan ip range

For example:

- msr("10.2.2.10","10.2.2.98","Standard");
- msr("10.2.2.10","10.2.2.98","OCP");

Parameters:

startip - the ip of range start
endip - the ip of range end
rackType - Type of Rack

---

ntp

```
public void ntp(java.lang.String bmcUserName,
                java.lang.String bmcUserPasswd,
                java.lang.String[] rawCommandData,
                java.lang.String ip)
```

Set NTP for single node

For Example:

- GbtUtilityAPI gbt = new GbtUtilityAPI();
- String rawCode = {"0x01","0x00","0x0a","0x00","0x81","0x06","0x0f","0x1c"};
- gbt.ntp("admin,"password",rawCode, "10.2.2.100");

Parameters:

bmcUserName - BMC user name
bmcUserpasswd - BMC user password
rawCommandData - String[] type IPMI raw command code {"OpMode","Mode","Update Frequency","Update Frequency","IP1_1","IP1_2","IP1_3","IP1_4",}
ip - BMC IP

---

ntp

```
public void ntp(java.lang.String bmcUserName,
                java.lang.String bmcUserPasswd,
                java.lang.String[] rawCommandData,
                java.lang.String startip,
                java.lang.String endip,
                java.lang.String rackType)
```

Set NTP for ip range

For Example:

- GbtUtilityAPI gbt = new GbtUtilityAPI();
- String rawCode = {"0x01","0x00","0x0a","0x00","0x81","0x06","0x0f","0x1c"};
- gbt.ntp("admin,"password","rawCode, "10.2.2.1", "10.2.2.100" , "Standard");

Parameters:

rawCommandData - String[] type IPMI raw command code {"OpMode","Mode","Update Frequency","Update Frequency","IP1_1","IP1_2","IP1_3","IP1_4",}
startip - start BMC IP for scan option
endip - end BMC IP for scan option
rackType - rack type of node server default using "Standard"

---

ntp

```
public void ntp(java.lang.String[] rawCommandData)
```

Set NTP by using nodelist

For Example:

- GbtUtilityAPI gbt = new GbtUtilityAPI();
- String rawCode = {"0x01","0x00","0x0a","0x00","0x81","0x06","0x0f","0x1c"};
- gbt.ntp(rawCode);

Parameters:

rawCommandData - String[] type IPMI raw command code {"OpMode","Mode","Update Frequency","Update Frequency","IP1_1","IP1_2","IP1_3","IP1_4",}

---

pef

```
public void pef(java.lang.String action,
                java.lang.String PET)
```

get platform event for single node

For example:

- pef("reboot","enable");
- pef("off","disable");

Parameters:

action - the action of pef
PET - generate platform event trap when node get platform event.

---

pef

```
public void pef(java.lang.String bmcUserName,
                java.lang.String bmcUserPasswd,
                java.lang.String action,
                java.lang.String ip,
                java.lang.String PET)
```

get platform event for single node

For example:

- pef("reboot","10.2.2.10","enable");
- pef("off","10.2.2.10","disable");

Parameters:

action - the action of pef
ip - the ip of node
PET - generate platform event trap when node get platform event.

---

pef

```
public void pef(java.lang.String bmcUserName,
                java.lang.String bmcUserPasswd,
                java.lang.String action,
                java.lang.String startip,
                java.lang.String endip,
                java.lang.String rackType,
                java.lang.String PET)
```

get platform event by scan ip range

For example:

- pef("reboot","10.2.2.10","10.2.2.98","Standard","enable");
- pef("off","10.2.2.10","10.2.2.98","OCP","disable");

Parameters:

action - the action of pef
startip - the ip of range start
endip - the ip of range end
rackType - Type of Rack
PET - generate platform event trap when node get platform event.

---

power

```
public void power()
```

Show target power reading by using Nodelist

For example:

- power();

---

power

```
public void power(java.lang.String bmcUserName,
                  java.lang.String bmcUserPasswd,
                  java.lang.String ip)
```

Show target power reading for single node

For example:

- power("10.2.2.10");

Parameters:

ip - the ip of node

---

power

```
public void power(java.lang.String bmcUserName,
                  java.lang.String bmcUserPasswd,
                  java.lang.String startip,
                  java.lang.String endip,
                  java.lang.String rackType)
```

Show target power reading by scan ip range

For example:

- power("10.2.2.10","10.2.2.98","Standard");
- power("10.2.2.10","10.2.2.98","OCP");

Parameters:

startip - the ip of range start
endip - the ip of range end
rackType - Type of Rack

---

psu

```
public void psu(java.lang.String action)
```

Show 2 byte status and status diagnosis by using Nodelist

For example:

- psu("ff");
- psu("b2");

Parameters:

action - input bytes

---

psu

```
public void psu(java.lang.String bmcUserName,
                java.lang.String bmcUserPasswd,
                java.lang.String action,
                java.lang.String ip)
```

Show 2 byte status and status diagnosis for single node

For example:

- psu("ff","10.2.2.10");
- psu("b2","10.2.2.10");

Parameters:

action - input bytes
ip - the ip of node

---

psu

```
public void psu(java.lang.String bmcUserName,
                java.lang.String bmcUserPasswd,
                java.lang.String action,
                java.lang.String startip,
                java.lang.String endip,
                java.lang.String rackType)
```

Show 2 byte status and status diagnosis by scan ip range

For example:

- psu("ff","10.2.2.10","10.2.2.98","Standard");
- psu("b2","10.2.2.10","10.2.2.98","OCP");

Parameters:

action - input bytes
startip - the ip of range start
endip - the ip of range end
rackType - Type of Rack

---

rackex

```
public void rackex(GbtDefinition.rackexDefinition action)
```

Update rmc firmware, Dump rack log include rmc, control board and node and Unlock abnormal rmc with several BMC IPs, which are contained on NodeList.log file. If NodeList.log file does not exist or you wanted to change IP range. Please using Scan method create a new NodeList.log file before using this method. Or you can just using another argument method. {@link GbtUtilityAPI#rackex(rackexDefinition, String, String, String)}. The following usage hints of this method will guide you how to use.

For Example:

- GbtUtilityAPI gbt = new GbtUtilityAPI();
- gbt.rackex(rackexDefinition.update.setUpdateArgs(String Firmware, boolean auto, String MACList_path));
- gbt.rackex(rackexDefinition.dump);
- gbt.rackex(rackexDefinition.unlock);

Parameters:

action - rackexDefinition type of load default option (tag has BIOS BMC and Cancel)

---

rackex

```
public void rackex(java.lang.String bmcUserName,
                   java.lang.String bmcUserPasswd,
                   GbtDefinition.rackexDefinition action,
                   java.lang.String ip)
```

Update rmc firmware, Dump rack log include rmc, control board and node and Unlock abnormal rmc with single BMC IP. The following usage hints of this method will guide you how to use.

For Example:

- GbtUtilityAPI gbt = new GbtUtilityAPI();
- gbt.rackex(rackexDefinition.update.setUpdateArgs(String Firmware, boolean auto, String MACList_path), "10.2.2.1");
- gbt.rackex(rackexDefinition.dump, "10.2.2.1");
- gbt.rackex(rackexDefinition.unlock, "10.2.2.1");

Parameters:

action - rackexDefinition type of load default option (tag has BIOS BMC and Cancel)
ip - single BMC IP

---

rackex

```
public void rackex(java.lang.String bmcUserName,
                   java.lang.String bmcUserPasswd,
                   GbtDefinition.rackexDefinition action,
                   java.lang.String startip,
                   java.lang.String endip,
                   java.lang.String rackType)
```

Update rmc firmware, Dump rack log include rmc, control board and node and Unlock abnormal rmc with several BMC IPs, which are contained on NodeList.log file. The scan option will scan IPs between startip and endip. After scan these IP will be written into NodeList.log. The following usage hints of this method will guide you how to use.

For Example:

- GbtUtilityAPI gbt = new GbtUtilityAPI();
- gbt.rackex(rackexDefinition.update.setUpdateArgs(String Firmware, boolean auto, String MACList_path),"10.2.2.1", "10.2.2.1", "10.2.2.100" , "Standard");
- gbt.rackex(rackexDefinition.dump);
- gbt.rackex(rackexDefinition.unlock);

Parameters:

action - rackexDefinition type of load default option (tag has BIOS BMC and Cancel)
startip - start BMC IP for scan option
endip - end BMC IP for scan option
rackType - rack type of node server default using "Standard"

---

rackst

```
public void rackst(GbtDefinition.rackstDefinition action)
```

Update rmc or fan board and Rack management only support RMC v1.0 with several BMC IPs, which are contained on NodeList.log file. If NodeList.log file does not exist or you wanted to change IP range. Please using Scan method create a new NodeList.log file before using this method. Or you can just using another argument method. {@link GbtUtilityAPI#rackst(rackstDefinition, String, String, String)}. The following usage hints of this method will guide you how to use.

For Example:

- GbtUtilityAPI gbt = new GbtUtilityAPI();
- gbt.rackst(rackstDefinition.update.setDevAndFW(GbtDevice.rmc), "updatefirmware");
- gbt.rackst(rackstDefinition.update.setDevAndFW(GbtDevice.fanboard), "updatefirmware");
- gbt.rackst(rackstDefinition.setRMCcommand("RMC command"));

Parameters:

action - rackstDefinition type of load default option (tag has BIOS BMC and Cancel)

---

rackst

```
public void rackst(GbtDefinition.rackstDefinition action,
                   java.lang.String ip)
```

Update rmc or fan board and Rack management only support RMC v1.0 with single BMC IP The following usage hints of this method will guide you how to use.

For Example:

- GbtUtilityAPI gbt = new GbtUtilityAPI();
- gbt.rackst(rackstDefinition.update.setDevAndFW(GbtDevice.rmc), "updatefirmware","10.2.2.1");
- gbt.rackst(rackstDefinition.update.setDevAndFW(GbtDevice.fanboard), "updatefirmware","10.2.2.1");
- gbt.rackst(rackstDefinition.setRMCcommand("RMC command"),"10.2.2.1");

Parameters:

action - rackstDefinition type of load default option (tag has BIOS BMC and Cancel)
ip - single BMC IP

---

rackst

```
public void rackst(GbtDefinition.rackstDefinition action,
                   java.lang.String startip,
                   java.lang.String endip,
                   java.lang.String rackType)
```

Update rmc or fan board and Rack management only support RMC v1.0 with several BMC IPs, which are contained on NodeList.log file. The scan option will scan IPs between startip and endip. After scan these IP will be written into NodeList.log. The following usage hints of this method will guide you how to use.

For Example:

- GbtUtilityAPI gbt = new GbtUtilityAPI();
- gbt.rackst(rackstDefinition.update.setDevAndFW(GbtDevice.rmc), "updatefirmware","10.2.2.1", "10.2.2.1", "10.2.2.100" , "Standard");
- gbt.rackst(rackstDefinition.update.setDevAndFW(GbtDevice.fanboard), "updatefirmware","10.2.2.1", "10.2.2.1", "10.2.2.100" , "Standard");
- gbt.rackst(rackstDefinition.setRMCcommand("RMC command"),"10.2.2.1", "10.2.2.1", "10.2.2.100" , "Standard");

Parameters:

action - loaddfDefinition type of load default option (tag has BIOS BMC and Cancel)
startip - start BMC IP for scan option
endip - end BMC IP for scan option
rackType - rack type of node server default using "Standard"

---

raw

```
public void raw(java.lang.String bmcUserName,
                java.lang.String bmcUserPasswd,
                java.lang.String[] rawCommandData,
                java.lang.String ip)
```

Send IPMI raw command to BMC with single IP. The execute message will be written into each IP's log file. Ex : 10.2.7.13_RawData.log. The following usage hints of this method will guide you how to use.

For Example:

- GbtUtilityAPI gbt = new GbtUtilityAPI();
- String rawCode = {"0x06", "0x01"};
- gbt.raw(rawCode, "10.2.2.1");

Parameters:

rawCommandData - String[] type IPMI raw command code {NetFn, Cmd, requestData}
ip - single BMC IP

---

raw

```
public void raw(java.lang.String bmcUserName,
                java.lang.String bmcUserPasswd,
                java.lang.String[] rawCommandData,
                java.lang.String startip,
                java.lang.String endip,
                java.lang.String rackType)
```

Send IPMI raw command to BMC with several IPs, which are contained on NodeList.log file. The scan option will scan IPs between startip and endip. After scan these IP will be written into NodeList.log. The execute message will be written into each IP's log file. Ex : 10.2.7.13_RawData.log. The following usage hints of this method will guide you how to use.

For Example:

- GbtUtilityAPI gbt = new GbtUtilityAPI();
- String rawCode = {"0x06", "0x01"};
- gbt.raw(rawCode, "10.2.2.1", "10.2.2.100" , "Standard");

Parameters:

rawCommandData - String[] type IPMI raw command code {NetFn, Cmd, requestData}
startip - start BMC IP for scan option
endip - end BMC IP for scan option
rackType - rack type of node server default using "Standard"

---

raw

```
public void raw(java.lang.String[] rawCommandData)
```

Send IPMI raw command to BMC with several IPs, which are contained on NodeList.log file. If NodeList.log file does not exist or you wanted to change IP range. Please using Scan method create a new NodeList.log file before using this method. Or you can just using another argument method. {@link GbtUtilityAPI#raw(String[], String, String, String)}. The execute message will be written into each IP's log file. Ex : 10.2.7.13_RawData.log. The following usage hints of this method will guide you how to use.

For Example:

- GbtUtilityAPI gbt = new GbtUtilityAPI();
- String rawCode = {"0x06", "0x01"};
- gbt.raw(rawCode);

Parameters:

rawCommandData - String[] type IPMI raw command code {NetFn, Cmd, requestData}

---

sdr

```
public void sdr()
```

show all sensor reading by using Nodelist

---

sdr

```
public void sdr(java.lang.String bmcUserName,
                java.lang.String bmcUserPasswd,
                java.lang.String ip)
```

show all sensor reading for single node

For example:

- sdr("10.2.2.10"");

Parameters:

ip - the ip of node

---

sdr

```
public void sdr(java.lang.String bmcUserName,
                java.lang.String bmcUserPasswd,
                java.lang.String startip,
                java.lang.String endip,
                java.lang.String rackType)
```

show all sensor reading by scan ip range

For example:

- sdr("10.2.2.10","10.2.2.98","Standard");
- sdr("10.2.2.10","10.2.2.98","OCP");

Parameters:

startip - the ip of range start
endip - the ip of range end
rackType - Type of Rack

---

sel

```
public void sel(java.lang.String action)
```

get/clear target node sensor event log by using Nodelist

For example:

- sel("get");
- sel("clear");

Parameters:

action - get/clear target node sensor event log

---

sel

```
public void sel(java.lang.String bmcUserName,
                java.lang.String bmcUserPasswd,
                java.lang.String action,
                java.lang.String ip)
```

get/clear target node sensor event log for single node

For example:

- sel("get","10.2.2.10");
- sel("clear","10.2.2.10");

Parameters:

action - get/clear target node sensor event log
ip - the ip of node

---

sel

```
public void sel(java.lang.String bmcUserName,
                java.lang.String bmcUserPasswd,
                java.lang.String action,
                java.lang.String startip,
                java.lang.String endip,
                java.lang.String rackType)
```

get/clear target node sensor event log by scan ip range

For example:

- sel("get","10.2.2.10","10.2.2.98","Standard");
- sel("clear","10.2.2.10","10.2.2.98","OCP");

Parameters:

action - get/clear target node sensor event log
startip - the ip of range start
endip - the ip of range end
rackType - Type of Rack

---

smbios

```
public void smbios()
```

Get smbios information with several BMC IPs, which are contained on NodeList.log file. If NodeList.log file does not exist or you wanted to change IP range. Please using Scan method create a new NodeList.log file before using this method. Or you can just using another argument method. {@link GbtUtilityAPI#smbios(String, String, String)}. The following usage hints of this method will guide you how to use.

For Example:

- GbtUtilityAPI gbt = new GbtUtilityAPI();
- gbt.smbios();

---

smbios

```
public void smbios(java.lang.String bmcUserName,
                   java.lang.String bmcUserPasswd,
                   java.lang.String ip)
```

Get smbios information with single BMC IP. The following usage hints of this method will guide you how to use.

For Example:

- GbtUtilityAPI gbt = new GbtUtilityAPI();
- gbt.smbios("10.2.2.1");

Parameters:

ip - single BMC IP

---

smbios

```
public void smbios(java.lang.String bmcUserName,
                   java.lang.String bmcUserPasswd,
                   java.lang.String startip,
                   java.lang.String endip,
                   java.lang.String rackType)
```

Get smbios information with several BMC IPs, which are contained on NodeList.log file. The scan option will scan IPs between startip and endip. After scan these IP will be written into NodeList.log. The following usage hints of this method will guide you how to use.

For Example:

- GbtUtilityAPI gbt = new GbtUtilityAPI();
- gbt.smbios("10.2.2.1", "10.2.2.1", "10.2.2.100" , "Standard");

Parameters:

startip - start BMC IP for scan option
endip - end BMC IP for scan option
rackType - rack type of node server default using "Standard"

---

trap

```
public void trap(java.lang.String action,
                 java.lang.String ip1,
                 java.lang.String ip2,
                 java.lang.String ip3,
                 java.lang.String ip4)
```

set ipv4/ipv6 trap to target node by using Nodelist

For example:

- trap("ipv4","ip1","ip2","ip3","ip4");
- trap("ipv6","ip1","ip2","ip3","ip4");

Parameters:

action - the action of trap
ip1 - use IP to enable ip1 ip, set 0 to disable ip1
ip2 - use IP to enable ip2 ip, set 0 to disable ip2
ip3 - use IP to enable ip3 ip, set 0 to disable ip3
ip4 - use IP to enable ip4 ip, set 0 to disable ip4

---

trap

```
public void trap(java.lang.String bmcUserName,
                 java.lang.String bmcUserPasswd,
                 java.lang.String action,
                 java.lang.String ip,
                 java.lang.String ip1,
                 java.lang.String ip2,
                 java.lang.String ip3,
                 java.lang.String ip4)
```

set ipv4/ipv6 trap to target node for single node

For example:

- trap("ipv4","10.2.2.10","ip1","ip2","ip3","ip4");
- trap("ipv6","2002:c706:1ef:1::1001","ip1","ip2","ip3","ip4");

Parameters:

action - the action of trap
ip - the ip of node
ip1 - use IP to enable ip1 ip, set 0 to disable ip1
ip2 - use IP to enable ip2 ip, set 0 to disable ip2
ip3 - use IP to enable ip3 ip, set 0 to disable ip3
ip4 - use IP to enable ip4 ip, set 0 to disable ip4

---

trap

```
public void trap(java.lang.String bmcUserName,
                 java.lang.String bmcUserPasswd,
                 java.lang.String action,
                 java.lang.String startip,
                 java.lang.String endip,
                 java.lang.String rackType,
                 java.lang.String ip1,
                 java.lang.String ip2,
                 java.lang.String ip3,
                 java.lang.String ip4)
```

set ipv4/ipv6 trap to target node by scan ip range

For example:

- trap("ipv4","10.2.2.10","10.2.2.98","Standard","ip1","ip2","ip3","ip4");
- trap("ipv6","2002:c706:1ef:1::1001","2002:c706:1ef:1::1010","OCP","ip1","ip2","ip3","ip4");

Parameters:

action - the action of trap
startip - the ip of range start
endip - the ip of range end
rackType - Type of Rack
ip1 - use IP to enable ip1 ip, set 0 to disable ip1
ip2 - use IP to enable ip2 ip, set 0 to disable ip2
ip3 - use IP to enable ip3 ip, set 0 to disable ip3
ip4 - use IP to enable ip4 ip, set 0 to disable ip4

update

`public void` **`update`**`(GbtDefinition.updateDefinition action)`

Update firmware utility of BMC, BIOS and CPLD with several BMC IPs, which are contained on NodeList.log file. If NodeList.log file does not exist or you wanted to change IP range. Please using Scan method create a new NodeList.log file before using this method. Or you can just using another argument method. {@link GbtUtilityAPI#update(updateDefinition, String, String, String)}. The following usage hints of this method will guide you how to use.

For Example:

- GbtUtilityAPI gbt = new GbtUtilityAPI();
- gbt.update(updateDefinition.updBIOS.setBIOSupdateArgs(String[] firmwares , int autoMode, String tftpPath, String tftpIP);
- gbt.update(updateDefinition.updBMC.setBMCupdateArgs(String[] firmwares , int autoMode, String tftpPath, String tftpIP);
- gbt.update(updateDefinition.updCPLD.setCPLDupdateArgs(GbtDevice.bpb, "FWfile.vme", "10.5.1.1", "/tftpboot"));
- gbt.update(updateDefinition.updCPLD.setCPLDupdateArgs(GbtDevice.mb, "FWfile.jbc", "10.5.1.1", "/tftpboot"));
- gbt.update(updateDefinition.updPSU.setPSUupdateArgs("FWfile.bin", "10.5.1.1", "/tftpboot"));

Parameters:

action - updateDefinition type of load default option (tag has updBIOS updBMC and updCPLD)

update

```
public void update(java.lang.String bmcUserName,
                   java.lang.String bmcUserPasswd,
                   GbtDefinition.updateDefinition action,
                   java.lang.String ip)
```

Update firmware utility of BMC, BIOS and CPLD with single BMC IP The following usage hints of this method will guide you how to use.

For Example:

- GbtUtilityAPI gbt = new GbtUtilityAPI();
- gbt.update(updateDefinition.updBIOS.setBIOSupdateArgs(String[] firmwares , int autoMode, String tftpPath, String tftpIP), "10.2.2.1");
- gbt.update(updateDefinition.updBMC.setBMCupdateArgs(String[] firmwares , int autoMode, String tftpPath, String tftpIP), "10.2.2.1");
- gbt.update(updateDefinition.updCPLD.setCPLDupdateArgs(GbtDevice.bpb, "FWfile.vme", "10.5.1.1", "/tftpboot"), "10.2.2.1");
- gbt.update(updateDefinition.updCPLD.setCPLDupdateArgs(GbtDevice.mb, "FWfile.jbc", "10.5.1.1", "/tftpboot"), "10.2.2.1");
- gbt.update(updateDefinition.updPSU.setPSUupdateArgs("FWfile.bin", "10.5.1.1", "/tftpboot"),"10.2.2.1");

Parameters:

action - updateDefinition type of load default option (tag has updBIOS updBMC and updCPLD)
ip - single BMC IP

update

```
public void update(java.lang.String bmcUserName,
                   java.lang.String bmcUserPasswd,
                   GbtDefinition.updateDefinition action,
                   java.lang.String startip,
                   java.lang.String endip,
                   java.lang.String rackType)
```

Update firmware utility of BMC, BIOS and CPLD with several BMC IPs, which are contained on NodeList.log file. The scan option will scan IPs between startip and endip. After scan these IP will be written into NodeList.log. The following usage hints of this method will guide you how to use.

For Example:

- GbtUtilityAPI gbt = new GbtUtilityAPI();
- gbt.update(updateDefinition.updBIOS.setBIOSupdateArgs(String[] firmwares , int autoMode, String tftpPath, String tftpIP), "10.2.2.1", "10.2.2.100" , "Standard");
- gbt.update(updateDefinition.updBMC.setBMCupdateArgs(String[] firmwares , int autoMode, String tftpPath, String tftpIP), "10.2.2.1", "10.2.2.100" , "Standard");
- gbt.update(updateDefinition.updCPLD.setCPLDupdateArgs(GbtDevice.bpb, "FWfile.vme", "10.5.1.1", "/tftpboot"), "10.2.2.1", "10.2.2.100" , "Standard");
- gbt.update(updateDefinition.updCPLD.setCPLDupdateArgs(GbtDevice.mb, "FWfile.jbc", "10.5.1.1", "/tftpboot"), "10.2.2.1", "10.2.2.100" , "Standard");
- gbt.update(updateDefinition.updPSU.setPSUupdateArgs("FWfile.bin", "10.5.1.1", "/tftpboot"), "10.2.2.1", "10.2.2.100" , "Standard");

Parameters:

action - updateDefinition type of load default option (tag has updBIOS updBMC and updCPLD)
startip - start BMC IP for scan option
endip - end BMC IP for scan option
rackType - rack type of node server default using "Standard"

userapp

```
public void userapp(GbtDefinition.userappDefinition action)
```

Set IPMI user account and list all users via BMC with several IPs, which are contained on NodeList.log file. If NodeList.log file does not exist or you wanted to change IP range. Please using Scan method create a new NodeList.log file before using this method. Or you can just using another argument method. {@link GbtUtilityAPI#userapp(userappDefinition, String, String, String)}. The "Set" and "List" option will write the log into each BMC IP's log file. Ex : 10.2.7.13_UserAccountSetRecords.log for "Set" option and 10.2.7.13_UserAccountListRecords.log for "List" option The following usage hints of this method will guide you how to use.

For Example:

Set option :

- GbtUtilityAPI gbt = new GbtUtilityAPI();
- gbt.userapp(userappDefinition.Set.setAccount(username, password, userid, priv));

List option :

- gbt.userapp(userappDefinition.List);

Parameters:

action - This Argument is a {@link userappDefinition} type userapp command, it contains two options {@link userappDefinition#Set} and {@link userappDefinition#List}.

---

userapp

```
public void userapp(java.lang.String bmcUserName,
                    java.lang.String bmcUserPasswd,
                    GbtDefinition.userappDefinition action,
                    java.lang.String ip)
```

Set IPMI user account and list all users via BMC with single IP, The "Set" and "List" option will write the log into each BMC IP's log file. Ex : 10.2.7.13_UserAccountSetRecords.log for "Set" option and 10.2.7.13_UserAccountListRecords.log for "List" option The following usage hints of this method will guide you how to use.

For Example:

Set option :

- GbtUtilityAPI gbt = new GbtUtilityAPI();
- gbt.userapp(userappDefinition.Set.setAccount(username, password, userid, priv), "10.2.2.1");

List option :

- gbt.userapp(userappDefinition.List, "10.2.2.1");

Parameters:

action - This Argument is a {@link userappDefinition} type userapp command, it contains two options {@link userappDefinition#Set} and {@link userappDefinition#List}.
ip - single BMC IP

userapp

```
public void userapp(java.lang.String bmcUserName,
                    java.lang.String bmcUserPasswd,
                    GbtDefinition.userappDefinition action,
                    java.lang.String startip,
                    java.lang.String endip,
                    java.lang.String rackType)
```

Set IPMI user account and list all users via BMC with several IPs, which are contained on NodeList.log file. The scan option will scan IPs between startip and endip. After scan these IP will be written into NodeList.log. The "Set" and "List" option will write the log into each BMC IP's log file. Ex : 10.2.7.13_UserAccountSetRecords.log for "Set" option and 10.2.7.13_UserAccountListRecords.log for "List" option The following usage hints of this method will guide you how to use.

GIGABYTE ™

For Example:

Set option :

- GbtUtilityAPI gbt = new GbtUtilityAPI();
- gbt.userapp(userappDefinition.Set.setAccount(username, password, userid, priv), "10.2.2.1", "10.2.2.100" , "Standard");

List option :

- gbt.userapp(userappDefinition.List, "10.2.2.1", "10.2.2.100" , "Standard");

Parameters:

action - This Argument is a {@link userappDefinition} type userapp command, it contains two options {@link userappDefinition#Set} and {@link userappDefinition#List}.
startip - start BMC IP for scan option
endip - end BMC IP for scan option
rackType - rack type of node server default using "Standard"

vmcli

```
public void vmcli(GbtDefinition.vmcliDefinition action)
```

Virtual media utility mount or umount vertual media via BMC with several IPs, which are contained on NodeList.log file. If NodeList.log file does not exist or you wanted to change IP range. Please using Scan method create a new NodeList.log file before using this method. Or you can just using another argument vmcli method ({@link GbtUtilityAPI#vmcli(vmcliDefinition, String, String, String) }). The execute message will be written into each IP's log file. Ex : 10.2.7.13_vmcliRecords.log. The following usage hints of this method will guide you how to use.

For Example:

Mount case :

- GbtUtilityAPI gbt = new GbtUtilityAPI();
- gbt.vmcli(vmcliDefinition.Mount.setVertualMediaDevices("media.iso", "floppy.img", ""));
- in method : setVertualMediaDevices(String isoname, String floppyImg, String removableDev). If you don not want to set device ex: floppyImg . please set "" string into argument

Umount case :

- gbt.vmcli(vmcliDefinition.Umount);

Parameters:

action - This Argument is a {@link vmcliDefinition} type vmcli command, it contains two options "Mount" and "Umount". When using "Mount" definition, please using "setVertualMediaDevices" method in {@link vmcliDefinition} to set vertual media file. It will store at {@link vmcliDefinition.Mount} with a static String[] form for this method using. "Umount" option without any arguments, it will umount vertual media, which had been mounted on server.

vmcli

```
public void vmcli(java.lang.String bmcUserName,
                  java.lang.String bmcUserPasswd,
                  GbtDefinition.vmcliDefinition action,
                  java.lang.String ip)
```

Virtual media utility mount or umount vertual media via BMC single IP. The execute message will be written into each IP's log file. Ex : 10.2.7.13_vmcliRecords.log. The following usage hints of this method will guide you how to use.

For Example:

Mount case :

- GbtUtilityAPI gbt = new GbtUtilityAPI();
- gbt.vmcli(vmcliDefinition.Mount.setVertualMediaDevices("media.iso", "floppy.img", ""),"10.2.2.1");
- in method : setVertualMediaDevices(String isoname, String floppyImg, String removableDev). If you don not want to set device ex: floppyImg . please set "" string into argument

Umount case :

- gbt.vmcli(vmcliDefinition.Umount,,"10.2.2.1");

Parameters:

action - This Argument is a {@link vmcliDefinition} type vmcli command, it contains two options "Mount" and "Umount". When using "Mount" definition, please using "setVertualMediaDevices" method in {@link vmcliDefinition} to set vertual media file. It will store at {@link vmcliDefinition#Mount} with a static String[] form for this method using. "Umount" option without any arguments, it will umount vertual media, which had been mounted on server.
ip - single BMC IP

---

vmcli

```
public void vmcli(java.lang.String bmcUserName,
                  java.lang.String bmcUserPasswd,
                  GbtDefinition.vmcliDefinition action,
                  java.lang.String startip,
                  java.lang.String endip,
                  java.lang.String rackType)
```

Virtual media utility mount or umount vertual media via BMC with several IPs, which are contained on NodeList.log file. The scan option will scan IPs between startip and endip. After scan these IP will be written into NodeList.log. The execute message will be written into each IP's log file. Ex : 10.2.7.13_vmcliRecords.log. The following usage hints of this method will guide you how to use.

For Example:

Mount case :

- GbtUtilityAPI gbt = new GbtUtilityAPI();
- gbt.vmcli(vmcliDefinition.Mount.setVertualMediaDevices("media.iso", "floppy.img", ""), "10.2.2.1", "10.2.2.100" , "Standard");
- in method : setVertualMediaDevices(String isoname, String floppyImg, String removableDev). If you don not want to set device ex: floppyImg . please set "" string into argument

Umount case :

- gbt.vmcli(vmcliDefinition.Umount, "10.2.2.1", "10.2.2.100" , "Standard");

Parameters:

action - This Argument is a {@link vmcliDefinition} type vmcli command, it contains two options "Mount" and "Umount". When using "Mount" definition, please using "setVertualMediaDevices" method in {@link vmcliDefinition} to set vertual media file. It will store at {@link vmcliDefinition#Mount} with a static String[] form for this method using. "Umount" option without any arguments, it will umount vertual media, which had been mounted on server.
startip - start BMC IP for scan option
endip - end BMC IP for scan option
rackType - rack type of node server default using "Standard"