

ASUS UEFI BIOS Windows Management Interface

Revision 2.03

Revision History

Data	Revision	Description
January 2022	2.03	Revise the description of SetFactoryTestMode.
July 2021	2.02	Only Update Version.
January 2021	2.01A	Add clear Password command.
January 2021	2.01	1.Add CHASIS FAN2 and AUX0 FAN Support to GetFanPolicy/ SetFanPolicy 2. Add fan source info to GetFanPolicy/ SetFanPolicy
November 2020	2.00	This version only support RockLake and upon.
August 2017	1.03	1, Add the following functions: <i>CheckPassword</i> <i>GetOptionData</i> <i>SetOptionData</i>
July 2017	1.02	Add the following functions. <i>SetDeviceControl</i> <i>GetFanPolicy</i> <i>SetFanPolicy</i> <i>GetManualFanCurve</i> <i>SetManualFanCurve</i>
April 2017	1.01	Add some function.
October 2014	1.00	First Release

Contents

1 Overview	4
1.1 Intended Audience	6
1.2 ASUS WMI Details	6
2 UEFI Background	7
2.1 Boot Order	7
2.2 Fast Boot	8
3 Function Description	9
3.1 GetLastError	9
3.2 GetBootOrder	10
3.3 SetBootOrder	10
3.4 GetBootOptionName	11
3.5 IsLegacyBootGroup	12
3.6 GetLegacyBootOrder	12
3.7 SetLegacyBootOrder	13

3.8 GetLegacyBootOptionName	14
3.9 SetBootOrderByGroup	15
3.10 SetAdminPassword	16
3.11 SetUserPassword	17
3.12 LoadDefaults	17
3.13 SetFastBoot	18
3.14 SetOptionDefault	19
3.15 SetFactoryTestMode	20
3.16 SetDeviceControl.....	21
3.17 GetFanPolicy.....	22
3.18 SetFanPolicy	23
3.19 GetManualFanCurve	25
3.20 SetManualFanCurve.....	27
3.21 CheckPassword	28
3.22 GetOptionData	28
3.23 SetOptionData.....	29
4 VB Script Samples.....	30
4.1 GetLastError.vbs	30
4.2 GetBootOrder.vbs.....	31
4.3 SetLegacyBootOrder.vbs	32

1 Overview

More and more requirement about IT administrators how to easier manage the client computer, about HW management, SW management, BIOS management and more. In this document, ASUS BIOS WMI interface provides a simplified way to manage client computer's BIOS related function.

Windows Management Instrumentation (WMI) is the infrastructure for management data and operations on Windows-based operating systems. You can write WMI scripts or applications to automate administrative tasks on remote computers but WMI also supplies management data to other parts of the operating

system and products, for example System Center Operations Manager, formerly Microsoft Operations Manager (MOM), or Windows Remote Management (WinRM).

ASUS WMI interface follows Microsoft's WMI White Paper, and implement it for specific function of ASUS products, including password control, boot priority control, POST flow control, BIOS SETUP item control and so on, which could be helpful for customer's system management and factory manufacture flow.

Related documents:

Windows Management Instrumentation

[https://msdn.microsoft.com/en-us/library/aa394582\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/aa394582(v=vs.85).aspx)

Windows Instrumentation: WMI and ACPI

[https://msdn.microsoft.com/en-us/library/windows/hardware/dn614028\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/hardware/dn614028(v=vs.85).aspx)

1.1 Intended Audience

This document is intended for the following readers:

IT administrators who will manage client computer.

SW developers who develop computer management software on ASUS platform.

1.2 ASUS WMI Details

ASUS WMI is implemented under the following scope.

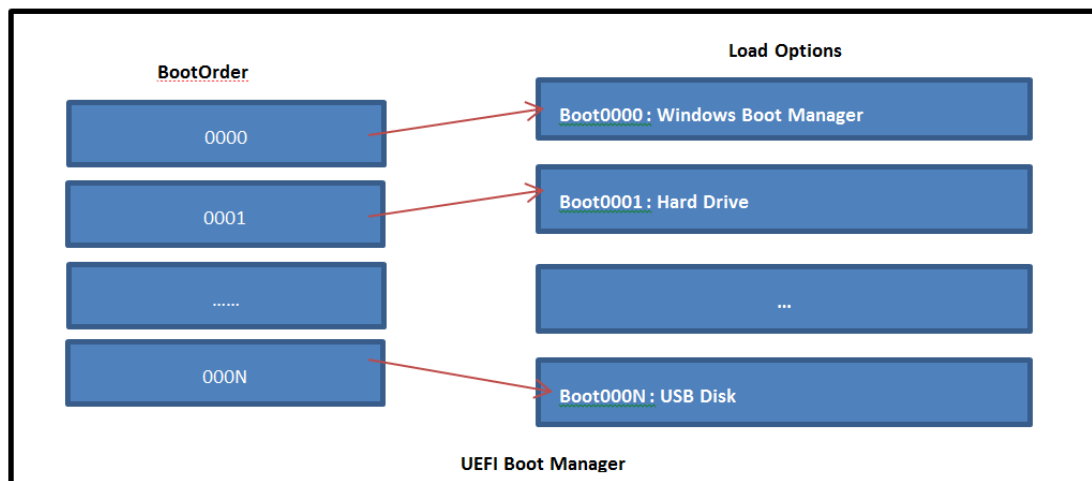
Namespace : **\root\wmi**
Class : **ASUSManagement**
InstanceName : **ACPI\PNP0C14\ASUSWMI_0**

2 UEFI Background

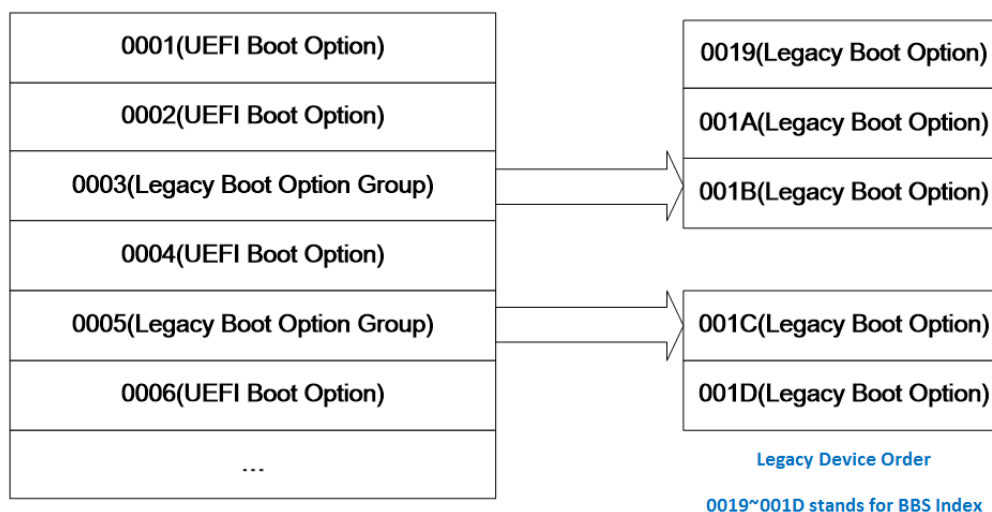
2.1 Boot Order

As for UEFI, each boot option is described by a Load Option, which resides in a **Boot####** variable, and all the **Boot####** are organized by a **BootOrder** variable. (#### is replaced by a unique option number in printable hexadecimal representation using the digits 0–9, and the upper case versions of the characters A–F (0000–FFFF)).

UEFI BIOS boot manger will read **BootOrder** variable and try to boot from **Boot####** one by one. In the following diagram, namely 0000 -> 0001 -> ... -> 000N.



But in fact, as legacy boot options need to be compatible, it's more complex. Legacy boot options of the same type will be grouped into one simple Load option.



2.2 Fast Boot

In order to speed up POST time, ASUS UEFI BIOS implement Fast Boot POST flow, which initialize minimum required system devices, and boot to previous Boot Option directly.

But under some conditions, for example, if user wants to change boot order, or user wants system fully initialized, it needs to disable Fast Boot to make new setting take effects.

3 Function Description

In this chapter, it will introduce ASUS detail WMI interfaces, and give examples for each of it.

Please notice that every time after the method execution, you need to call **GetLastError()** function to retrieve the method's execution result. If the method executes successful, the **GetLastError()** will return 0, otherwise failed. The detail about **GetLastError()**, please see 2.1.

3.1 GetLastError

Summary

Retrieve previous method's execution result.

Prototype

```
void GetLastError(
    [out] uint32 ErrorCode
);
```

Parameters

ErrorCode 0 means successful, otherwise failed.

Related Definitions

None

Description

The GetLastError function retrieves previous method's execution result. Please notice that every time after the method execution, you need to call **GetLastError()** function to retrieve the method's execution result. (Windows WMI API always return success even if BIOS interface failure, that's why you need you call GetLastError to get ErrorCode from BIOS.)

Example

```
>wmic /namespace:\\root\wmi path ASUSManagement where
InstanceName="ACPI\PNP0C14\ASUSWMI_0" call GetLastError
```

3.2 GetBootOrder

Summary

Get current Boot Order.

Prototype

```
void GetBootOrder(
    [out] uint32 ErrorCode,
    [out] String BootOrder
);
```

Parameters

<i>ErrorCode</i>	0 means successful, otherwise failed.
<i>BootOrder</i>	Current UEFI boot order.

Related Definitions

None

Description

The GetBootOrder function retrieves current boot order. BootOrder is data array of WORD size. For example, 0001000300040002 stands for Boot0001 -> Boot0003 -> Boot0004 -> Boot0002.

To retrieve the name of Boot####, please call GetBootOrderName with parameter "####".

Example

```
> wmic /namespace:\\root\wmi path ASUSManagement where
    InstanceName="ACPI\PNP0C14\ASUSWMI_0" call GetBootOrder
```

3.3 SetBootOrder

Summary

Set new Boot Order.

Prototype

```
void SetBootOrder(
    [in] String BootOrder,
    [out] uint32 ErrorCode
);
```

Parameters

<i>BootOrder</i>	New UEFI boot order.
<i>ErrorCode</i>	0 means successful, otherwise failed.

Related Definitions

None

Description

The SetBootOrder function update a new boot order.

Example

```
> wmic /namespace:\\root\wmi path ASUSManagement where
  InstanceName="ACPI\PNP0C14\ASUSWMI_0" call SetBootOrder
  BootOrder=000200030001
```

3.4 GetBootOptionName

Summary

Get current Boot Order.

Prototype

```
void
GetBootOptionName(
    [in] uint16 Handle,
    [out] uint32 ErrorCode,
    [out] String BootOptionName
);
```

Parameters

Handle	Boot Option Index, namely "####" of Boot####.
ErrorCode	0 means successful, otherwise failed.
BootOptionName	Name string of this Boot option.

Related Definitions

None

Description

The GetBootOptionName function retrieves the name string of this Boot Option indicated by Handle. (If handle equals 0003, GetBootOptionName return the name of Boot Option "Boot0003" into BootOptionName)

Example

```
> wmic /namespace:\\root\wmi path ASUSManagement where
  InstanceName="ACPI\PNP0C14\ASUSWMI_0" call GetBootOrderName
  Handle=0003
```

3.5 IsLegacyBootGroup

Summary

Check if this Boot Option is a legacy boot device group.

Prototype

```
void IsLegacyBootGroup(
    [in] uint16 Handle,
    [out] uint32 ErrorCode,
    [out] boolean IsLegacyGroup
);
```

Parameters

Handle	Boot Option Index, namely "#####" of Boot####.
ErrorCode	0 means successful, otherwise failed.
IsLegacyGroup	True for legacy boot device group, false for not.

Related Definitions

None

Description

If Boot#### is a legacy (BBS) boot option, UEFI BIOS will create a dummy Boot Option (in fact a group header) and a **LegacyDevOrder** variable to describe its detail information. That's why we implement IsLegacyBootGroup function to check if it's a legacy Boot Option. If it's legacy(BBS) Boot Option, Boot Option name could be get by GetLegacyBootOptionName. If it's UEFI Boot Option(not legacy), Boot Option name could be get by GetBootOptionName.

Example

```
> wmic /namespace:\\root\wmi path ASUSManagement where
  InstanceName="ACPI\PNP0C14\ASUSWMI_0" call IsLegacyBootGroup
  Handle=0002
```

3.6 GetLegacyBootOrder

Summary

Get Legacy device Boot Order.

Prototype

```
void GetLegacyBootOrder(
    [in] uint16 Handle,
```

```

[out] uint32 ErrorCode,
[out] String LegacyBootOrder
);

```

Parameters

Handle Boot Option Index, namely "####" of Boot####.

ErrorCode 0 means successful, otherwise failed.

LegacyBootOrder Legacy Boot Order of this device type indicated by Boot####.

Related Definitions

LegacyBootOrder is of the following format. 0019/001A... stands for BBS Index.

Boot0001	→	HDD	Length	0019	001A	001B
Boot0003	→	CD-ROM	Length	001C	001D	

Description

GetLegacyBootOrder retrieves the LegacyDevOrder whose type is same as Boot#### indicated by Handle. You may need to use this function to get the Legacy Device Order Index(e.g. 0019), and then get its detail name string by GetLegacyBootOptionName.

Example

```

> wmic /namespace:\root\wmi path ASUSManagement where
InstanceName="ACPI\PNP0C14\ASUSWMI_0" call GetLegacyBootOrder
Handle=0003

```

3.7 SetLegacyBootOrder

Summary

Set Legacy device Boot Order.

Prototype

```

void SetLegacyBootOrder(
    [in] uint16 Handle,
    [in] String LegacyBootOrder,
    [out] uint32 ErrorCode,
);

```

Parameters

Handle Boot Option Index, namely "####" of Boot####.

LegacyBootOrder New Legacy Boot Order of this device type indicated by

Boot####.

ErrorCode 0 means successful, otherwise failed.

Related Definitions

None

Description

SetLegacyBootOrder update the LegacyDevOrder whose type is same as Boot#### indicated by Handle.

Example

```
> wmic /namespace:\\root\wmi path ASUSManagement where
InstanceName="ACPI\PNP0C14\ASUSWMI_0" call SetLegacyBootOrder
Handle=0003 LegacyBootOrder=001A0019001B
```

3.8 GetLegacyBootOptionName

Summary

Get current Boot Order.

Prototype

```
void
GetLegacyBootOptionName(
    [in] uint16 Handle,
    [out] uint32 ErrorCode,
    [out] String BootOptionName
);
```

Parameters

Handle BBS Index.
ErrorCode 0 means successful, otherwise failed.
BootOptionName Name string of this Boot option.

Related Definitions

None

Description

The GetLegacyBootOptionName function retrieves the name string of this Legacy Boot Option indicated by Handle(BBS Index). (If handle equals 001A, GetBootOptionName return the name of Boot Option with BBS Index 001A)

Example

```
> wmic /namespace:\\root\wmi path ASUSManagement where
```

*InstanceName="ACPI\\PNP0C14\\ASUSWMI_0" call GetLegacyBootOptionName
Handle=001A*

3.9 SetBootOrderByGroup

Summary

Set Boot Order By Group.

Prototype

```
void SetBootOrderByGroup(
    [in] String BootOrderByGroup,
    [out] uint32 ErrorCode
);
```

Parameters

<i>BootOrderByGroup</i>	New Boot Order described by a formatted string.
<i>ErrorCode</i>	0 means successful, otherwise failed.

Related Definitions

BootOrderByGroup is combined by the following types.

"uH"	-	UEFI Hard Disk (e.g. Windows Boot Manager)
"uO"	-	UEFI ODD/CD-ROM
"uL"	-	UEFI LAN (e.g. UEFI IPV4/IPV6)
"uR"	-	UEFI Removable Device(e.g UEFI USB boot disk)
"H"	-	Legacy Hard Disk
"O"	-	Legacy ODD/CD-ROM
"L"	-	Legacy LAN (e.g. PXE)
"R"	-	Legacy Removable Device(e.g USB boot disk)

If BootOrderByGroup is "uHuOuLHOLuRR", new Boot Order is "UEFI HDD -> UEFI ODD -> UEFI LAN -> HDD -> ODD -> LAN -> UEFI Removable -> Removable".

If BootOrderByGroup is "uHH", new Boot Order is "UEFI HDD -> HDD", with other Boot devices disabled.

Description

The SetBootOrderByGroup function enroll a new Boot Order into BIOS, and make it works during next boot. It's an easy way to adjust boot priority next boot, even new devices plug-in.

Example

```
> wmic /namespace:\\root\\wmi path ASUSManagement where  
InstanceName="ACPI\\PNP0C14\\ASUSWMI_0" call SetBootOrderByGroup
```

BootOrderByGroup=[uHuOuLuRHOLR](#)

3.10 SetAdminPassword

Summary

Set SETUP administrator password.

Prototype

```
void SetAdminPassword(
    [in] String CurrentPassword,
    [in] String NewPassword,
    [out] uint32 ErrorCode
);
```

Parameters

CurrentPassword	Current administrator password.
NewPassword	New administrator password.
ErrorCode	0 means successful, otherwise failed.

Related Definitions

None

Description

SetAdminPassword enroll an administrator password into UEFI BIOS. Please make sure CurrentPassword is correct, or the operation will not be successful and GetLastError return an error code. If current administrator password is not installed, please use NULL instead. If administrator password installed, use "CL" command for clear the administrator password. The password length is limited to 3 ~ 20 characters.

Example

```
> wmic /namespace:\\root\wmi path ASUSManagement where
  InstanceName="ACPI\PNP0C14\ASUSWMI_0" call SetAdminPassword
  CurrentPassword=NULL NewPassword=12345678
  Clear Current Password call SetAdminPassword
  CurrentPassword=12345678 Command=CL
```

3.11 SetUserPassword

Summary

Set SETUP user password.

Prototype

```
void SetUserPassword(
    [in] String CurrentPassword,
    [in] String NewPassword,
    [out] uint32 ErrorCode
);
```

Parameters

CurrentPassword	Current administrator password.
NewPassword	New administrator password.
ErrorCode	0 means successful, otherwise failed.

Related Definitions

None

Description

SetUserPassword enroll a user password into UEFI BIOS. Please make sure CurrentPassword is correct, or the operation will not be successful and GetLastError return an error code. If current user password is not installed, please use NULL instead. If user password installed, use "CL" command for clear the user password. The password length is limited to 3 ~ 20 characters.

Example

```
> wmic /namespace:\\root\wmi path ASUSManagement where
InstanceName="ACPI\PNP0C14\ASUSWMI_0" call SetUserPassword
CurrentPassword=123 NewPassword=456
Clear New Password call SetUserPassword CurrentPassword=456 Command=CL
```

3.12 LoadDefaults

Summary

Set SETUP user password.

Prototype

```
void LoadDefaults([out] uint32 ErrorCode);
```

Parameters

ErrorCode Load Defaults return codes, 0 for success.

Related Definitions

None

Description

LoadDefaults function will load SETUP default settings during next boot. Please notice that not all the setting be reset, some critical options will be kept for security reason such as Secure Boot Mode.

Example

*> wmic /namespace:\\root\wmi path ASUSManagement where
InstanceName="ACPI\PNP0C14\ASUSWMI_0" call LoadDefaults*

3.13 SetFastBoot

Summary

Set SETUP user password.

Prototype

```
void SetFastBoot(
    [in] uint8 SetValue,
    [out] uint32 ErrorCode
);
```

Parameters

SetValue 1 for enable, 0 for disable.

ErrorCode 0 means successful, otherwise failed.

Related Definitions

None

Description

SetFastBoot function will enable/disable UEFI fast boot feature. This is important if you want system full-initialized or re-initialized. Please notice that SetFastBoot only change the current setting of UEFI fast boot, with its default setting un-changed. If you want to change its default value, please call SetOptionDefault.

Example

*> wmic /namespace:\\root\wmi path ASUSManagement where
InstanceName="ACPI\PNP0C14\ASUSWMI_0" call SetFastBoot SetValue=0*

3.14 SetOptionDefault

Summary

Set SETUP Option default value.

Prototype

```
void SetOptionDefault(
    [in] String OptionName,
    [in] uint32 Value,
    [out] uint32 ErrorCode
);
```

Parameters

OptionName SETUP Option Name.

Value Value of this SETUP Option.

ErrorCode 0 means successful, otherwise failed.

Related Definitions

OptionName varies on different platform or different models, as their SETUP options are not the same. For safe, please contact the corresponding BIOS engineer for option names.

There could be a table which describes Option names as below.

1.1 CPU Configuration Group

Option Name	Description	Value
VT	Intel (VMX) Virtualization Technology	0 – Disable 1 – Enable

1.2 System Agent (SA) Configuration

Option Name	Description	Value
EnableVtd	VT-d	0 – Disable 1 – Enable

Please notice do not add prefix for OptionName.

Description

SetOptionDefault function will change one of SETUP options' default value, and make system default configuration during next boot.

Example

```
> wmic /namespace:\root\wmi path ASUSManagement where
InstanceName="ACPI\PNP0C14\ASUSWMI_0" call SetOptionDefault
OptionName=FastBoot Value=0
```

3.15 SetFactoryTestMode

Summary

Set Factory Test Mode.

Prototype

```
void SetFactoryTestMode(  
    [out] uint32 ErrorCode  
);
```

Parameters

ErrorCode 0 means successful, otherwise failed.

Related Definitions

None

Description

ASUS Factory Test Mode enable some convenient BIOS setting for factory manufacture.(Such as : The system will not pause to wait for 'F1', when any error occurs.)

Example

```
> wmic /namespace:\\root\\wmi path ASUSManagement where  
InstanceName="ACPI\\PNP0C14\\ASUSWMI_0" call SetFactoryTestMode
```

3.16 SetDeviceControl

Summary

Set device function control.

Prototype

```
void SetDeviceControl(
    [in] String DeviceName,
    [in] uint32 Value,
    [out] uint32 ErrorCode
);
```

Parameters

DeviceName Device Name
 Value 0 for disable, 1 for enable
 ErrorCode 0 means successful, otherwise failed.

Related Definitions

DeviceName supported:

“COM1”

“COM2”

“LPT”

Description

SetDeviceControl provide an interface for device on/off control. It works after system reboot.

Example

```
> wmic /namespace:\\root\wmi path ASUSManagement where
InstanceName="ACPI\PNP0C14\ASUSWMI_0" call SetDeviceControl DeviceName=COM1 Value=0
```

3.17 GetFanPolicy

Summary

Get fan policy by fan type.

Prototype

```
void GetFanPolicy(
    [in] uint8 FanType,
    [out] uint32 ErrorCode,
    [out] String Mode,
    [out] uint32 LowLimit,
    [out] String Profile,
    [out] String Source
);
```

Parameters

<i>FanType</i>	Fan Type, 0 for CPU FAN, 1 for CHASIS FAN, 2 for CHASIS FAN2, 3 for AUX0 FAN.
<i>ErrorCode</i>	0 means successful, otherwise failed.
<i>Mode</i>	Fan working mode.
<i>LowLimit</i>	Low limit of fan tachometer for warning.
<i>Profile</i>	Fan curve profile.
<i>Source</i>	Fan Source, but CPU FAN not support.

Related Definitions

Supported fan mode is as below:

Supported FAN Mode	Notes
AUTO	DC/PWM by auto detect, need HW support
DC	DC Mode, need HW support
PWM	PWM Mode, need HW support

Supported fan profile is as below:

Supported FAN Mode	Notes
STANDARD	
SILENT	Lower fan RPM will be applied
TURBO	Higher fan RPM will be applied
MANUAL	User specified FAN curve will be applied
FULLSPEED	

Supported fan Source is as below:

Supported FAN Mode	Notes
CPU	Fan Source is CPU

MOTHERBOARD	Fan Source is MotherBoard
MULTIPLESOURCES	Fan Source is Multiple Sources

Description

GetFanPolicy retrieves the current fan setting indicated by *FanType*.

Example

> *wmic /namespace:\root\wmi path ASUSManagement where InstanceName="ACPI\PNP0C14\ASUSWMI_0" call GetFanPolicy FanType= [0](#)*

3.18 SetFanPolicy

Summary

Set fan policy to specified fan.

Prototype

```
void SetFanPolicy(
    [in] uint8 FanType,
    [in] String Mode,
    [in] uint32 LowLimit,
    [in] String Profile,
    [in] String Source,
    [out] uint32 ErrorCode
);
```

Parameters

<i>FanType</i>	Fan Type, 0 for CPU FAN, 1 for CHASIS FAN, 2 for CHASIS FAN2, 3 for AUX0 FAN.
<i>Mode</i>	Fan working mode.
<i>LowLimit</i>	Low limit of fan tachometer for warning.
<i>Profile</i>	Fan curve profile.
<i>Source</i>	Fan Source, but CPU FAN not support.
<i>ErrorCode</i>	0 means successful, otherwise failed.

Related Definitions

Mode, *Profile* and *Source* definition is the same as them in GetFanPolicy.

Please notice that setting an unsupported fan mode or profile to system will lead to unknown behavior. So, please confirm the HW/SETUP/manual first.

Description

SetFanPolicy will update the current fan setting indicated by *FanType*. The new fan policy will work after system reboot.

Example

```
> wmic /namespace:\\root\wmi path ASUSManagement where  
InstanceName="ACPI\PNP0C14\ASUSWMI_0" call SetFanPolicy FanType=0 Mode =  
AUTO LowLimit = 300 Profile = TURBO
```

```
> wmic /namespace:\\root\wmi path ASUSManagement where  
InstanceName="ACPI\PNP0C14\ASUSWMI_0" call SetFanPolicy FanType=1 Mode =  
AUTO LowLimit = 300 Profile = TURBO Source = MOTHERBOARD
```

3.19 GetManualFanCurve

Summary

Get manual profile setting by fan type and mode.

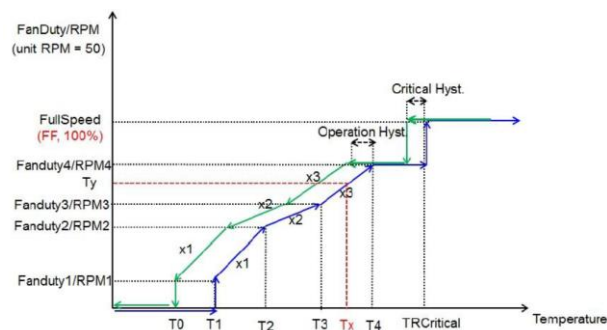
Prototype

```
void GetManualFanCurve (
    [in] uint8 FanType,
    [in] String Mode,
    [out] uint32 ErrorCode,
    [out] uint8 LowTemp,
    [out] uint8 LowDuty,
    [out] uint8 MidTemp,
    [out] uint8 MidDuty,
    [out] uint8 HighTemp,
    [out] uint8 HighDuty
);
```

Parameters

<i>FanType</i>	Fan Type, 0 for CPU FAN, 1 for CHASIS FAN.
<i>Mode</i>	Fan working mode.
<i>ErrorCode</i>	0 means successful, otherwise failed.
<i>LowTemp</i>	Lower Temperature.
<i>LowDuty</i>	Min Duty Cycle(%).
<i>MidTemp</i>	Middle temperature.
<i>MidDuty</i>	Middle Duty Cycle(%).
<i>HighTemp</i>	Max temperature.
<i>HighDuty</i>	Max Duty Cycle(%).

Related Definitions



For the above diagram, *LowTemp* stands for *T1*, *MidTemp* stands for *T3*, and *HighTemp* stands for *TRcritical*.

Description

GetManualFanCurve will retrieve the current fan curve indicated by *FanType* and *Mode*. Fan working at different Mode may have different fan curve.

Example

```
> wmic /namespace:\\root\wmi path ASUSManagement where  
InstanceName="ACPI\PNP0C14\ASUSWMI_0" call GetManualFanCurve FanType=0  
Mode = PWM
```

3.20 SetManualFanCurve

Summary

Set manual profile setting by fan type and mode.

Prototype

```
void SetManualFanCurve (
    [in] uint8 FanType,
    [in] String Mode,
    [out] uint32 ErrorCode,
    [out] uint8 LowTemp,
    [out] uint8 LowDuty,
    [out] uint8 MidTemp,
    [out] uint8 MidDuty,
    [out] uint8 HighTemp,
    [out] uint8 HighDuty
);
```

Parameters

<i>FanType</i>	Fan Type, 0 for CPU FAN, 1 for CHASIS FAN.
<i>Mode</i>	Fan working mode.
<i>LowTemp</i>	Lower Temperature.
<i>LowDuty</i>	Min Duty Cycle(%).
<i>MidTemp</i>	Middle temperature.
<i>MidDuty</i>	Middle Duty Cycle(%).
<i>HighTemp</i>	Max temperature.
<i>HighDuty</i>	Max Duty Cycle(%).
<i>ErrorCode</i>	0 means successful, otherwise failed.

Related Definitions

See GetManualFanCurve.

Description

SetManualFanCurve will update the current fan curve indicated by *FanType* and *Mode*. The new fan curve only works when fan profile is select to MANUAL mode, and will works after system reboot.

Fan working at different Mode may have different fan curve.

Example

```
>wmic /namespace:\\root\wmi path ASUSManagement where
InstanceName="ACPI\PNP0C14\ASUSWMI_0" call SetManualFanCurve FanType=0 Mode=DC
LowTemp=20 LowDuty=30 MidTemp=40 MidDuty=50 HighTemp=60 HighDuty=100
```

3.21 CheckPassword

Summary

Do SETUP password check action to unlock SETUP related WMI functions.

Prototype

```
void CheckPassword (
    [in] String CurrentPassword,
    [out] uint32 ErrorCode,
);
```

Parameters

<i>CurrentPassword</i>	Current administrator password for SETUP.
<i>ErrorCode</i>	0 means successful, otherwise failed.

Related Definitions

None

Description

CheckPassword function does password provision for SETUP administrator password.

Please notes that from ASUS WMI v1.03, all functions return *ErrorCode* 15 for “access denied” until *CheckPassword* function been called and return success.(Except password related functions)

Example

```
>wmic /namespace:\\root\wmi path ASUSManagement where
InstanceName="ACPI\\PNP0C14\\ASUSWMI_0" call CheckPassword CurrentPassword=123456
```

3.22 GetOptionData

Summary

Get SETUP Option current value.

Prototype

```
void GetOptionData (
    [in] String OptionName,
    [out] uint32 ErrorCode,
    [out] uint32 Value
);
```

Parameters

<i>OptionName</i>	SETUP Option Name.
<i>ErrorCode</i>	0 means successful, otherwise failed.

Value Current value of this SETUP Option.

Related Definitions

OptionName varies on different platform or different models, as their SETUP options are not the same. For safe, please contact the corresponding BIOS engineer for option names. (Please refer 3.14 *SetOptionDefaults* for detail.)

Description

GetOptionData function will retrieve one of SETUP options' current value.

Example

```
> wmic /namespace:\root\wmi path ASUSManagement where
InstanceName="ACPI\PNP0C14\ASUSWMI_0" call GetOptionData
OptionName=FastBoot
```

3.23 SetOptionData

Summary

Set SETUP Option with new value.

Prototype

```
void SetOptionData (
    [in] String OptionName,
    [in] uint32 Value
    [out] uint32 ErrorCode
);
```

Parameters

OptionName	SETUP Option Name.
Value	New value of this SETUP Option to be set.
ErrorCode	0 means successful, otherwise failed.

Related Definitions

OptionName varies on different platform or different models, as their SETUP options are not the same. For safe, please contact the corresponding BIOS engineer for option names. (Please refer 3.14 *SetOptionDefaults* for detail.)

Description

SetOptionData function will set one of SETUP options to a new value.

Example

```
> wmic /namespace:\root\wmi path ASUSManagement where
InstanceName="ACPI\PNP0C14\ASUSWMI_0" call SetOptionData
OptionName=FastBoot Value=0
```

4 VB Script Samples

4.1 GetLastError.vbs

```
strComputer = "."
Set objWMIService = GetObject("winmgmts:\\\" & strComputer & "\\root\\WMI")
' Obtain an instance of the the class
' using a key property value.
Set objShare =
objWMIService.Get("ASUSManagement.InstanceName='ACPI\\PNP0C14\\ASUSWMI_0'")

' no InParameters to define

' Execute the method and obtain the return status.
' The OutParameters object in objOutParams
' is created by the provider.
Set objOutParams =
objWMIService.ExecMethod("ASUSManagement.InstanceName='ACPI\\PNP0C14\\ASU
SWMI_0'", "GetLastError")

' List OutParams
Wscript.Echo "Out Parameters: "
Wscript.echo "ErrorCode: " & objOutParams.ErrorCode
```

4.2 GetBootOrder.vbs

```

strComputer = "."
Set objWMIService = GetObject("winmgmts:\\." & strComputer & "\root\WMI")
' Obtain an instance of the the class
' using a key property value.
Set objShare =
objWMIService.Get("ASUSManagement.InstanceName='ACPI\PNP0C14\ASUSWMI_0'")

' no InParameters to define

' Execute the method and obtain the return status.
' The OutParameters object in objOutParams
' is created by the provider.
Set objOutParams =
objWMIService.ExecMethod("ASUSManagement.InstanceName='ACPI\PNP0C14\ASU
SWMI_0'", "GetBootOrder")

' List OutParams
Wscript.Echo "Out Parameters: "
Wscript.Echo "BootOrder: " & objOutParams.BootOrder
Wscript.echo "ErrorCode: " & objOutParams.ErrorCode

```

3.3 SetLegacyBootOrder.vbs

```

strComputer = "."
Set objWMIService = GetObject("winmgmts:\\." & strComputer & "\root\WMI")
' Obtain an instance of the the class
' using a key property value.
Set objShare =
objWMIService.Get("ASUSManagement.InstanceName='ACPI\PNP0C14\ASUSWMI_0'")

' Obtain an InParameters object specific
' to the method.
Set objInParam = objShare.Methods_("SetLegacyBootOrder")._
    inParameters.SpawnInstance_()

' Add the input parameters.
objInParam.Properties_.Item("Handle") = 0001
objInParam.Properties_.Item("LegacyBootOrder") = "0019001A"

' Execute the method and obtain the return status.
' The OutParameters object in objOutParams
' is created by the provider.
Set objOutParams =
objWMIService.ExecMethod("ASUSManagement.InstanceName='ACPI\PNP0C14\ASU
SWMI_0'", "SetLegacyBootOrder", objInParam)

' List OutParams
Wscript.echo "ErrorCode: " & objOutParams.ErrorCode

```