TOSHIBA BIOS WMI Interface Guide

Version 2013, rev 1.1
Check Network access sharing and security model......................... 28
Check Firewall settings........................................................................... 29
Some general information about remote access................................. 29
Some general infos about access issues in combination with UAC .. 29
Special note for Windows 8 ................................................................. 29

A Visual Basic script to set a password on a remote computer .......... 30
B Sample scripts in PowerShell ................................................................. 35
  Some general notes on PowerShell ....................................................... 35
  Read all BIOS settings and output to the console ............................... 35
  Write a single BIOS item ................................................................. 35
  Save current BIOS settings to a file.................................................... 37
  Reads saved BIOS settings from a file and writes it back to BIOS ...... 38
  Set or Change BIOS Passwords ......................................................... 40
# List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 1</td>
<td>BIOS User Password</td>
<td>4</td>
</tr>
<tr>
<td>Table 2</td>
<td>BIOS Supervisor Password</td>
<td>5</td>
</tr>
<tr>
<td>Table 3</td>
<td>HDD Password</td>
<td>5</td>
</tr>
<tr>
<td>Table 4</td>
<td>Queries</td>
<td>7</td>
</tr>
<tr>
<td>Table 5</td>
<td>SMBIOS items with written property</td>
<td>9</td>
</tr>
<tr>
<td>Table 6</td>
<td>Queries Overview</td>
<td>9</td>
</tr>
<tr>
<td>Table 7</td>
<td>Method Overview</td>
<td>12</td>
</tr>
<tr>
<td>Table 8</td>
<td>Method Details</td>
<td>12</td>
</tr>
<tr>
<td>Table 9</td>
<td>Method Return Value</td>
<td>18</td>
</tr>
<tr>
<td>Table 10</td>
<td>Asset Tag BIOS item</td>
<td>19</td>
</tr>
<tr>
<td>Table 11</td>
<td>BIOS Settings</td>
<td>22</td>
</tr>
<tr>
<td>Table 12</td>
<td>Settings for the supervisor</td>
<td>26</td>
</tr>
</tbody>
</table>
Copyright, Disclaimer and Trademarks

Copyright

© 2013 by TOSHIBA Corporation. All rights reserved. Under the copyright laws, this manual cannot be reproduced in any form without the prior written permission of TOSHIBA. No patent liability is assumed, with respect to the use of the information contained herein.


Copyright authority for music, movies, computer programs, databases and other intellectual property covered by copyright laws belongs to the author or to the copyright owner. Copyrighted material can be reproduced only for personal use or use within the home. Any other use beyond that stipulated above (including conversion to digital format, alteration, transfer of copied material and distribution on a network) without the permission of the copyright owner is a violation of copyright or author's rights and is subject to civil damages or criminal action. Please comply with copyright laws in making any reproduction from this manual.

Disclaimer

This manual has been validated and reviewed for accuracy. The instructions and descriptions it contains are accurate for your computer at the time of this manual’s production. However, succeeding computers’ BIOS and manuals are subject to change without notice. TOSHIBA assumes no liability for damages incurred directly or indirectly from errors, omissions or discrepancies between the computers’ BIOS and the manual.

Trademarks

Intel and Intel vPro are trademarks or registered trademarks of Intel Corporation.

Windows, Microsoft and Windows logo are registered trademarks of Microsoft Corporation.

Target groups

This document is intended for IT administrators, IT specialists and service engineers that need to develop solutions for changing or controlling the TOSHIBA BIOS settings through the Windows Management Instrumentation (WMI) interface. The manual guides you through the features of Toshiba BIOS and exemplifies the usage of the WMI interface with script samples.

A deeper understanding of BIOS, PCs, Networking, WMI and Visual Basic script language is a prerequisite to reading this manual.
General precautions for changing BIOS settings

Be careful when you change the BIOS settings. If certain BIOS settings are not correctly configured, it is possible that:

- Some features or devices may not function properly.
- Computer or system boot failure occurs, possibly resulting in loss of data.

The BIOS Setup Screen can be accessed by pressing the F2 key when the TOSHIBA logo appears at boot time. If changes made to the BIOS result in system malfunction or undesired system performance, enter the BIOS again and press F9 to load Setup Defaults, and then press F10 to save and exit BIOS.

TOSHIBA Support

If you require any additional help using your computer or if you are having problems operating the computer, you may need to contact TOSHIBA for additional technical assistance.

Before you call

Some problems you experience may be related to software or the operating system, it is important to investigate other sources of assistance first. Before contacting TOSHIBA, try the following:

- Review troubleshooting sections in the documentation for software and peripheral devices.
- If a problem occurs when you are running software applications, consult the software documentation for troubleshooting suggestions. Call the software company's technical support for assistance.
- Consult the dealer you purchased your computer and/or software from.

Where to write

If you are still unable to solve the problem and suspect that it is hardware related, write to TOSHIBA at the location listed in the accompanying warranty booklet or visit http://www.toshiba.co.jp/worldwide/index.html on the Internet.
Toshiba BIOS and the WMI Interface

Overview

IT administrators try to find easy and quick solutions to manage the settings of the client computers’ BIOS. The Toshiba WMI interface offers a simple way to access the BIOS.

The Toshiba WMI interface enables the administrator to read and write all BIOS settings, reset the BIOS to factory settings, set and change passwords and modify the boot order.

Windows Management Instrumentation (WMI)

Windows Management Instrumentation (WMI) is in most Windows® operating systems included by default. It contains a wide range of functions:

- Start a process on a remote computer
- Schedule a process to be run on specific days at specific times
- Reboot a computer remotely
- Get a list of applications that are installed on a local computer or a remote computer
- Query the Windows event logs on a local computer or a remote computer

The Toshiba BIOS WMI provides additional functions. It facilitates the administration significantly.

The following illustration shows how the BIOS is controlled by WMI.

Figure 1 BIOS via WMI
Structure of the Toshiba BIOS

The Toshiba BIOS is divided into different areas – a general area with most BIOS settings intended for supervisors and a system management BIOS area (SMBIOS). As long as no supervisor password is set the general area and some fields of the SMBIOS area can be set freely. This is the default mode at the time the computer ships.

Once a supervisor password is set on the computer, which has to be done physically on each machine, the BIOS behavior changes. The BIOS will now require an authentication to modify any field values. Also a separate area of the BIOS will now be accessible. The supervisor will now have access to configure several security and boot related features.

The recommended scenario in a managed IT landscape is that the administrators set the supervisor password to restrict user access to the BIOS. This will prevent the users to modify any BIOS settings in an undesired way.

Passwords

The BIOS holds a number of passwords to restrict access either to the BIOS itself or to the hardware. Following is an explanation about each password and the area they protect. The passwords can partially be set, modified or deleted through the WMI interface either locally on a client or remotely over the network.

BIOS User Password

When the BIOS User Password is set, the user has to enter this password to access the BIOS through the Setup Utility. The Setup Utility can be accessed when pressing F2 at boot time.

The following table shows a state diagram of the User Password’s behavior.

Table 1  BIOS User Password

<table>
<thead>
<tr>
<th>Current Status</th>
<th>Action</th>
<th>Arguments</th>
<th>1st</th>
<th>2nd (Old)</th>
<th>3rd (New)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not Registered</td>
<td>Register</td>
<td>UserPassword</td>
<td>blank</td>
<td>PW</td>
<td></td>
</tr>
<tr>
<td>Registered</td>
<td>Change</td>
<td>UserPassword</td>
<td>PW</td>
<td>PW</td>
<td></td>
</tr>
<tr>
<td>Registered</td>
<td>Delete</td>
<td>UserPassword</td>
<td>PW</td>
<td>blank</td>
<td></td>
</tr>
</tbody>
</table>

For example: if a User Password is not set, it will be registered when calling SetPassword with “UserPassword” as first argument, a blank as second argument and a password as third argument. Once the User Password is registered, it can be either changed or deleted depending on the third argument in the call.

Examples on how to set the password through the WMI Interface will be given later in this chapter.

Note: When setting, changing or deleting passwords though the WMI Interface, the password(s) delivered as arguments need to be encoded. The encoded password can be generated by accessing the following website: https://www.biospw.com/tsb/encoder/
**BIOS Supervisor Password**

When the BIOS Supervisor Password is set, the BIOS will now require authentication to modify any BIOS setting. Access to several security configurations and boot related features will now be available. These features include enable/disable certain built-in components as well as restrict specific boot devices.

**Table 2  BIOS Supervisor Password**

<table>
<thead>
<tr>
<th>Current Status</th>
<th>Action</th>
<th>Arguments</th>
<th>1st</th>
<th>2nd (Old)</th>
<th>3rd (New)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not Registered</td>
<td>N/A</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Registered</td>
<td>Change</td>
<td>SupervisorPassword</td>
<td>PW</td>
<td>PW</td>
<td></td>
</tr>
<tr>
<td>Registered</td>
<td>Delete</td>
<td>SupervisorPassword</td>
<td>PW</td>
<td>blank</td>
<td></td>
</tr>
</tbody>
</table>

The Supervisor Password cannot be set initially over the WMI Interface. It has to be set manually on the client first. It has to be set either manually on the client first or by separate toll that can only be executed locally. Once the Supervisor Password is set, it can be changed or deleted via WMI interface.

If you need a tool to locally set the Supervisor Password, please contact your Toshiba representative.

**Note:** The password(s) delivered as arguments need to be encoded. The encoded password can be generated by accessing the following website: [https://www.biospw.com/tsb/encoder/](https://www.biospw.com/tsb/encoder/)

**HDD Password**

The HDD Password restricts the access to the HDD. There are two levels of passwords – a master password and a user password. The BIOS WMI Interface contains methods to set these passwords which are set physically onto the HDD.

When a HDD password is set, the HDD will require a valid password at boot time. When properly authenticated, the HDD can be accessed.

**Note:** The HDD Password locks the HDD and is not stored in the BIOS. A locked HDD cannot be accessed even if it is installed in another computer. The contents on a common HDD are not encrypted when a HDD password is set. However, this may be different on Self Encrypting Drives (SED).

**Table 3  HDD Password**

<table>
<thead>
<tr>
<th>Current Status</th>
<th>Action</th>
<th>Arguments</th>
<th>1st</th>
<th>2nd (Old)</th>
<th>3rd (New)</th>
<th>4th (Old)</th>
<th>5th (New)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not Registered</td>
<td>Not Registered</td>
<td>Register Master and User</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Registered</td>
<td>Not Registered</td>
<td>N/A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Not Registered</td>
<td>Registered</td>
<td>N/A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Registered</td>
<td>Registered</td>
<td>Change Only Master HDD password</td>
<td></td>
<td>Master PW</td>
<td>Master PW</td>
<td>blank</td>
<td>blank</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Change Only User HDD</td>
<td></td>
<td>blank</td>
<td>blank</td>
<td>User</td>
<td>User</td>
</tr>
<tr>
<td>password</td>
<td>PW</td>
<td>PW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------------</td>
<td>------</td>
<td>------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delete Master HDD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>password</td>
<td>Master</td>
<td>blank</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N/A</td>
<td>blank</td>
<td>blank</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Registered</th>
<th>Registered</th>
<th>N/A</th>
<th>UserOnlyHDDPassword</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not Registered</td>
<td>Not Registered</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Registered</td>
<td>Registered</td>
<td>Change User HDD password</td>
<td>User</td>
</tr>
<tr>
<td>Not Registered</td>
<td>Registered</td>
<td>Change User HDD password</td>
<td>User</td>
</tr>
<tr>
<td>Registered</td>
<td>Registered</td>
<td>Change User HDD password</td>
<td>User</td>
</tr>
</tbody>
</table>

**Note:** The password(s) delivered as arguments need to be encoded. The encoded password can be generated by accessing the following website: [https://www.biospw.com/tsb/encoder/](https://www.biospw.com/tsb/encoder/)
This chapter contains details on the WMI implementation for configuring BIOS settings. The queries can be used to find settings and their values. The methods are used to set or change settings.

## Configuring the BIOS settings

The following interface accesses the Toshiba BIOS settings.

**Namespace:** “\root\WMI”

**Base Class:** “ToshibaBIOSElement”

### Queries

<table>
<thead>
<tr>
<th>Class Name</th>
<th>Type</th>
<th>Return types</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>QueryTosIfVersion</td>
<td>Query</td>
<td>&quot;Value&quot;</td>
<td>&quot;1.00&quot;</td>
</tr>
<tr>
<td>QueryBiosSettings</td>
<td>Query</td>
<td>&quot;Item,Attribute,Value&quot;</td>
<td>&quot;WakeUpOnLAN,RW,Enabled&quot;</td>
</tr>
<tr>
<td>QueryBiosSettingsParameter</td>
<td>Query</td>
<td>&quot;Item,Value,...&quot;</td>
<td>&quot;WakeUpOnLAN,Disable,Enabled&quot;</td>
</tr>
<tr>
<td>QueryBiosItems</td>
<td>Query</td>
<td>&quot;item&quot;</td>
<td>&quot;WakeOnLAN&quot;</td>
</tr>
<tr>
<td>QuerySecurityPolicies</td>
<td>Query</td>
<td>&quot;item,Value&quot;</td>
<td>&quot;DeviceUSB,Enabled&quot;</td>
</tr>
<tr>
<td>QuerySecurityPoliciesParameter</td>
<td>Query</td>
<td>&quot;item,Value,...&quot;</td>
<td>&quot;DeviceUSB,Disable,Enabled&quot;</td>
</tr>
<tr>
<td>QuerySecurityPolicyItems</td>
<td>Query</td>
<td>&quot;item&quot;</td>
<td>&quot;DeviceUSB&quot;</td>
</tr>
<tr>
<td>QueryPasswordStatus</td>
<td>Query</td>
<td>&quot;item,Value&quot;</td>
<td>&quot;UserPassword,Registered&quot;</td>
</tr>
<tr>
<td>QueryPasswordItems</td>
<td>Query</td>
<td>&quot;item&quot;</td>
<td>&quot;UserPassword&quot;</td>
</tr>
<tr>
<td>QuerySmbiosStrings</td>
<td>Query</td>
<td>&quot;item,Value&quot;</td>
<td>&quot;BoardAssetTag,DEFAULT&quot;</td>
</tr>
<tr>
<td>QuerySmbiosItems</td>
<td>Query</td>
<td>&quot;item&quot;</td>
<td>&quot;BoardAssetTag&quot;</td>
</tr>
</tbody>
</table>

Note that all types returned are of type “String”. If there are several values returned, these will be separated by a comma (,). The return type “Attribute” refers to the access rights of a specific BIOS Setting Item – RW (Read Write), RO (Read Only), WO (Write Only).

Items and Values are case sensitive strings. If you want to address a certain BIOS setting, the name and value need to be spelled correctly. It is recommended to use queries to find out the correct spelling and possible values for that setting.
Using the queries

Here is a VB example on using the QueryBiosSettings Query to find out all Bios Settings and their values.

'Sample VBScript: List all BIOS settings on the local computer
'command line: cscript.exe ListAllBiosSettings.vbs

On Error Resume Next
Dim objWMIService, objItem, colItems, strComputer, strSetting, strItem, strValue

'define variables
strComputer = "." 'replace your computer name or leave "." as default value

'connect to WMI
Set objWMIService = GetObject("winmgmts:\" & strComputer & "\root\WMI")

If Err.Number <> 0 Then
WScript.Echo "Unable to connect to WMI service: " & Hex(Err.Number) & "." & New Line
WScript.Quit
End If

'exectues a WQL query
Set colItems = objWMIService.ExecQuery("Select * from QueryBiosSettings")

For Each objItem in colItems
If Len(objItem.CurrentSetting) > 0 Then
'return value contains elements, each separated by comma.
strSetting = ObjItem.CurrentSetting
strItem = Left(strSetting, InStr(strSetting, ",")) - 1
strValue = Mid(strSetting, InStr(strSetting, ",") + 1, 256)
WScript.Echo strItem + " = " + strValue
End If
Next

WScript.Quit

In a similar manner one can use the "QueryBiosSettingsParameter" query to find out possible parameters for each BIOS Setting:

Set colItems = objWMIService.ExecQuery("Select * from QueryBiosSettingsParameter")
This is quite a useful query to understand the possible values for each setting. If you need only to find the name of a setting, the query “QueryBiosItems” will be the appropriate one.

Three queries are designed to access the Security Policy settings. Once a Supervisor Password is set physically on a computer, a supervisor can enable and disable hardware or boot media and set various security related parameter to restrict an ordinary user. If authenticated as supervisor during boot up, the restrictions can be bypassed. These queries can be used to read the current security policy settings.

- **QuerySecurityPolicyItems** returns the names of the security policies, for instance “DeviceUSB”
- **QuerySecurityPolicies** returns the security policies and settings
- **QuerySecurityPoliciesParameter** returns the possible settings for each security policy

These two queries access the System management BIOS. The system management BIOS contain information about the manufacturer of the computer, the product name or serial number

- **QuerySmbiosItems** returns the names of the items in SMBIOS
- **QuerySmbiosStrings** returns the names and the values of the items in SMBIOS

There are two items in SMBIOS that can be written: **BoardAssetTag** and **ChassisAssetTagNumber**. These fields can be used for keeping track of hardware.

Each field can hold a maximum of 63 characters. If exceeded when setting, an error is returned.

**Table 5  SMBIOS items with written property**

<table>
<thead>
<tr>
<th>SMBIOS item</th>
<th>Item name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type2 Asset Tag</td>
<td>BoardAssetTag</td>
</tr>
<tr>
<td>Type3 Asset Tag Number</td>
<td>ChassisAssetTagNumber</td>
</tr>
</tbody>
</table>

The password queries return the names and status of the passwords. Please read the previous chapter on the different passwords and what they are used for.

- **QueryPasswordItems** returns the names of the passwords
- **QueryPasswordStatus** returns the name and the registration status of the password

The **QueryTosIfVersion** query returns the version of the Toshiba WMI Interface

Here is an overview of the queries and return values:

**Table 6  Queries Overview**

<table>
<thead>
<tr>
<th>Use this query...</th>
<th>Return structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>QueryTosIfVersion</td>
<td>To get the version of Toshiba WMI interface</td>
</tr>
<tr>
<td>Item</td>
<td>Version</td>
</tr>
<tr>
<td>QueryBiosSettings</td>
<td>To get the information of BIOS SETUP</td>
</tr>
<tr>
<td>Item</td>
<td>The name of BIOS SETUP</td>
</tr>
<tr>
<td>Attribute</td>
<td>The Read/Write attribute of BIOS SETUP</td>
</tr>
<tr>
<td>Value</td>
<td>The value of the item</td>
</tr>
</tbody>
</table>
### Using a query on a remote computer

The following script reads the BIOS settings and their parameters on a remote computer.

```vbs
On Error Resume Next
Dim objSWbemLocator, objWMIService, objItem, colItems, strComputer, strUserName, strPassword, strSetting, strItem, strValue

If WScript.Arguments.Count <> 3 Then
    WScript.Echo "Usage: cscript.exe ListAllBiosSettingsParameterRemote.vbs [ComputerName|IPAdress] [UserName] [Password]"
    WScript.Quit
End If
```

**Using a query on a remote computer**

<table>
<thead>
<tr>
<th>Query</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QueryBiosSettingsParameter</td>
<td>To get the acceptable settings of BIOS SETUP</td>
</tr>
<tr>
<td>Item</td>
<td>The name of BIOS SETUP</td>
</tr>
<tr>
<td>Value 1, 2, 3, …</td>
<td>The value(s) of the item</td>
</tr>
<tr>
<td>QueryBiosItems</td>
<td>To get the item names of BIOS SETUP</td>
</tr>
<tr>
<td>Item</td>
<td>The name of BIOS SETUP</td>
</tr>
<tr>
<td>QuerySecurityPolicies</td>
<td>To get the information of User Policy</td>
</tr>
<tr>
<td>Item</td>
<td>The name of User Policy</td>
</tr>
<tr>
<td>Value</td>
<td>The value of the item</td>
</tr>
<tr>
<td>QuerySecurityPoliciesParameter</td>
<td>To get the acceptable settings of User Policy</td>
</tr>
<tr>
<td>Item</td>
<td>The name of User Policy</td>
</tr>
<tr>
<td>Value 1, 2, 3, …</td>
<td>The value(s) of the item</td>
</tr>
<tr>
<td>QuerySecurityPolicyItems</td>
<td>To get the item names of User Policy</td>
</tr>
<tr>
<td>Item</td>
<td>The name of User Policy</td>
</tr>
<tr>
<td>QueryPasswordStatus</td>
<td>To get the registration status of passwords</td>
</tr>
<tr>
<td>Item</td>
<td>The password name</td>
</tr>
<tr>
<td>Value</td>
<td>The registration status of the password</td>
</tr>
<tr>
<td>QueryPasswordItems</td>
<td>To get the item names of passwords</td>
</tr>
<tr>
<td>Item</td>
<td>The password name</td>
</tr>
<tr>
<td>QuerySmbiosStrings</td>
<td>To get the SMBIOS strings</td>
</tr>
<tr>
<td>Item</td>
<td>The SMBIOS structure name of the string</td>
</tr>
<tr>
<td>Value</td>
<td>The string</td>
</tr>
<tr>
<td>QuerySmbiosItems</td>
<td>To get the structure names of SMBIOS</td>
</tr>
<tr>
<td>Item</td>
<td>The SMBIOS structure name of the string</td>
</tr>
</tbody>
</table>
'define variables
strComputer = WScript.Arguments(0)  'computer name or computer's IP address
strUserName = WScript.Arguments(1)
strPassword = WScript.Arguments(2)

wbemImpersonationLevelImpersonate = 3
wbemAuthenticationLevelPktPrivacy = 6

'get the locator object
Set objSWbemLocator = CreateObject("WbemScripting.SWbemLocator")

'get the service object from the remote server
Set objWMIService = objSWbemLocator.ConnectServer(strComputer, "root\WMI", strUserName, strPassword)

If Err.Number <> 0 Then
  WScript.Echo "Unable to connect to " & strComputer & ": " & Hex(Err.Number) & "."
  WScript.Quit
End If

objWMIService.Security_.ImpersonationLevel = wbemImpersonationLevelImpersonate
objWMIService.Security_.AuthenticationLevel = wbemAuthenticationLevelPktPrivacy

'executes a WQL query
Set colItems = objWMIService.ExecQuery("Select * from QueryBiosSettingsParameter")

For Each objItem in colItems
  If Len(objItem.CurrentSetting) > 0 Then
    'return value contains elements, each separated by comma.
    strSetting = objItem.ItemInformation
    If InStr(strSetting, ",") = 0 Then 'no comma found
      strItem = strSetting
      strValue = ""
    Else
      strItem = Left(strSetting, InStr(strSetting, ",") - 1)
      strValue = Mid(strSetting, InStr(strSetting, ",") + 1, 256)
    End If
    WScript.Echo strItem + ":" + strValue
  End If
Next
WScript.Quit

Methods

Methods are used to set and modify the BIOS settings. Here is an overview of the methods available.
### Table 7  Method Overview

<table>
<thead>
<tr>
<th>Class Name / Methods</th>
<th>Type</th>
<th>Instance name</th>
<th>Return/Parameter</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>ModeControl</td>
<td>Method</td>
<td>ACPI\PNP0C14\0_0</td>
<td>&quot;mode,svpw(*); &quot;</td>
<td>&quot;Start,1E302E; &quot;</td>
</tr>
<tr>
<td>BiosSetting</td>
<td>Method</td>
<td>ACPI\PNP0C14\0_0</td>
<td>&quot;Item;&quot;</td>
<td>&quot;WakeUpOnLAN,Enabled&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ACPI\PNP0C14\0_1</td>
<td>&quot;Execute;&quot;</td>
<td>&quot;Execute;&quot;</td>
</tr>
<tr>
<td>SetBiosSetting</td>
<td>Method</td>
<td>ACPI\PNP0C14\0_2</td>
<td>&quot;Item,Value;&quot;</td>
<td>&quot;WakeUpOnLAN,Disable;&quot;</td>
</tr>
<tr>
<td>SecurityPolicy</td>
<td>Method</td>
<td>ACPI\PNP0C14\0_0</td>
<td>&quot;Item;&quot;</td>
<td>&quot;DeviceUSB,Enabled&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ACPI\PNP0C14\0_1</td>
<td>&quot;Item,Value;&quot;</td>
<td>&quot;DeviceUSB,Disable;&quot;</td>
</tr>
<tr>
<td>Password</td>
<td>Method</td>
<td>ACPI\PNP0C14\0_0</td>
<td>&quot;Item;&quot;</td>
<td>&quot;UserPassword,Registered&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ACPI\PNP0C14\0_1</td>
<td>&quot;Item,OldPwd(<em>), NewPwd(</em>)&quot;</td>
<td>&quot;UserPassword,1E302E, 2D152C&quot;</td>
</tr>
<tr>
<td>SmBiosString</td>
<td>Method</td>
<td>ACPI\PNP0C14\0_0</td>
<td>&quot;Item;&quot;</td>
<td>&quot;BoardAssetTag,12345678&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ACPI\PNP0C14\0_1</td>
<td>&quot;Item,Value;&quot;</td>
<td>&quot;BoardAssetTag,12345678;&quot;</td>
</tr>
</tbody>
</table>

(*) The password(s) delivered as arguments need to be encoded. The encoded password can be generated by accessing the following website: [https://www.biospw.com/tsb/encoder/](https://www.biospw.com/tsb/encoder/)

In the following table you will find each Method with a short explanation and the parameter they require.

### Table 8  Method Details

<table>
<thead>
<tr>
<th>Method</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>SetConfigurationMode</td>
<td>If the supervisor password is registered, one needs to unlock the BIOS settings by sending Start and encoded Password. Start authenticates the supervisor. End closes the authentication.</td>
</tr>
<tr>
<td>GetBiosSetting</td>
<td>To get information about the BIOS Setting</td>
</tr>
<tr>
<td>LoadDefaultBiosSettings</td>
<td>Loads the default settings of the BIOS SETUP</td>
</tr>
<tr>
<td>SetBiosSetting</td>
<td>To set a specific BIOS Setting</td>
</tr>
<tr>
<td>GetSecurityPolicy</td>
<td>To get information of the User Policy</td>
</tr>
<tr>
<td>SetSecurityPolicy</td>
<td>To set the User Policy</td>
</tr>
</tbody>
</table>

Mode
- **Start**: Enter the mode where each BIOS Setting can be written when the supervisor password is registered
- **End**: Leave the mode where each BIOS Setting can be written when the supervisor password is registered

svpw
- Encoded supervisor password
### Method | Parameter
--- | ---
GetPasswordStatus | The value of the item
SetPassword | To get the registration status of the password
Item | The password name
OldPwd | To set a password
NewPwd | The name of passwords
Old encoded password | New encoded password
GetSmbiosString | To get the string of the SMBIOS
Item | The SMBIOS structure name
SetSmbiosString | To set the strings of SMBIOS
Item | The SMBIOS structure name of the string
Value | The string

### How to set a BIOS setting on a remote computer

The following code is a sample script on how to set a Bios setting on a remote computer. The script uses several methods and gives a representative example on how to use these methods. The scenario assumes that a supervisor password has been set (can only be set manually) on the remote computer.

You need an account with administrative rights on the remote computer to connect to. Otherwise you will not have enough privileges to modify a BIOS setting remotely.

Please save the following short script with the functions IsSupervisorPasswordRegistered and SetConfigurationMode into a file called "procedures.vbs". The main script will load these functions into memory and use them when required.

```vbs
Function IsSupervisorPasswordRegistered

    Function IsSupervisorPasswordRegistered (objWMIService)
    Dim isRegistered
    isRegistered = -1

    ''query the password status to check if the supervisor password is registered
    Set colItems = objWMIService.ExecQuery("Select * from Password where InstanceName='ACPI\PNP0C14\0_0'")

    For Each objItem in colItems
        'execute the method and obtain the return status
        objItem.GetPasswordStatus "SupervisorPassword;", strReturn

        ''return value contains two elements, each seperated by comma. e.g: "SupervisorPassword,Registered"
        strItem = Left(strReturn, InStr(strReturn, ",") - 1)
        strStatus = Mid(strReturn, InStr(strReturn, ",") + 1, 256)
    Next
    For Each objItem in colItems

    If strItem = "SupervisorPassword" And strStatus = "Registered" Then
        isRegistered = 1
    Next

    IsSupervisorPasswordRegistered = isRegistered

End Function
```
If strStatus = "Registered" Then
    isRegistered = 0
End If
Next

IsSupervisorPasswordRegistered = isRegistered

End Function

Function SetConfigurationMode(objWMIService, strInParamValue)
    Dim colItems, objItem
    'executes a WQL query
    Set colItems = objWMIService.ExecQuery("Select * from ModeControl where InstanceName='ACPI\PNP0C14\0_0'")
    For Each objItem in colItems
        'execute the method and obtain the return status
        objItem.SetConfigurationMode strInParamValue, strReturn
    Next
    SetConfigurationMode = strReturn
End Function

This is the main script on how set a single BIOS Setting on a remote computer.

'Sample VBScript: Set a single BIOS setting on a remote computer. Use this script if you have registered a supervisor password.

'command line: cscript.exe SetBiosConfigPasswordRemote.vbs [setting] [value] [scrambled SupervisorPassword] [ComputerName|IPAddress] [UserName] [Password]
'argument 1 : BIOS item name
'2 : the setting value you want to change
'3 : the scrambled supervisor Password
'4 : the IP, the FQDN, or the Computer name of the client PC you want to access
'5 : a username that has Administrator privileges on the client PC you want to access
'6 : password for the username that has Administrator privileges on the client PC you want to access

'declare application name
Dim strAppName
strAppName = "SetBiosConfigPasswordRemote"

On Error Resume Next
Dim objFSO, objFile, objWMIService, objItem, colItems, strComputer, strInParamValue, strReturn, strItem, strStatus,
strFileName, strSupervisorPassword, strParameter, strUserName, strPassword

' create Object to open the procedure file
strFileName = "procedures.vbs"
Set objFSO = CreateObject("Scripting.FileSystemObject")
Set objFile = objFSO.OpenTextFile(strFileName, 1)  ' 1 - for reading
Execute objFile.ReadAll()

' check input parameters
If WScript.Arguments.Count <> 6 Then
    WScript.Echo "Usage: cscript.exe SetBiosConfigPasswordRemote.vbs [setting] [value] [scrambled SupervisorPassword] [ComputerName|IPAddress] [UserName] [Password]"
    WScript.Quit
End If

'define variables
strInParamValue = WScript.Arguments(0) + ":" + WScript.Arguments(1) + ":" + strSupervisorPassword = WScript.Arguments(2)
strComputer = WScript.Arguments(3)  ' computer name or computer's IP address
strUserName = WScript.Argument(4)
strPassword = WScript.Arguments(5)

wbemImpersonationLevelImpersonate = 3
wbemAuthenticationLevelPktPrivacy = 6

' get the locator object
Set objSWbemLocator = CreateObject("WbemScripting.SWbemLocator")

' get the service object from the remote server
Set objWMIService = objSWbemLocator.ConnectServer(strComputer, "root\WMI", strUserName, strPassword)
If Err.Number <> 0 Then
    WScript.Echo "Unable to connect to " & strComputer & ": " & Hex(Err.Number) & ": "
    WScript.Quit
End If

objWMIService.Security_.ImpersonationLevel = wbemImpersonationLevelImpersonate
objWMIService.Security_.AuthenticationLevel = wbemAuthenticationLevelPktPrivacy

' check if the supervisor password is registered
strReturn = IsSupervisorPasswordRegistered(objWMIService)
If strReturn <> 0 Then
    WScript.Echo "You can not run this application if the supervisor password is not registered."
    WScript.Quit
End If

' authenticate with Supervisor privilege
strParameter = "Start," + strSupervisorPassword + ";"
strReturn = SetConfigurationMode(objWMIService, strParameter)
If strReturn <> 0 Then
    WScript.Echo "Supervisor password authentication failed. Error: " & GetErrMsg(Hex(strReturn))
    WScript.Quit
Else
    WScript.Echo "Supervisor password successfully authenticated."
End If

'executes a WQL query
Set colItems = objWMIService.ExecQuery("Select * from BiosSetting where InstanceName='ACPI\PNP0C14\0_0'")

' set single Bios setting
For Each objItem in colItems
    'execute the method and obtain the return status
    objItem.SetBiosSetting strInParamValue, strReturn
Next
WScript.Echo strAppName & ": " & GetErrMsg(Hex(strReturn))

'deauthenticate from supervisor mode
strParameter = "End," + strSupervisorPassword + ";"
strReturn = SetConfigurationMode(objWMIService, strParameter)
If strReturn <> 0 Then
    WScript.Echo "Supervisor password deauthentication failed. Error: " & GetErrMsg(Hex(strReturn))
    WScript.Quit
Else
    WScript.Echo "Supervisor password successfully deauthenticated."
End If
WScript.Quit

'convert an error code to a string
Function GetErrMsg(err)
    Dim strMsg
    Select Case err
        Case "0"
            strMsg = "The operation was successful."
        Case "8004100C"
            strMsg = "Feature or operation is not supported."
        Case "80041008"
            strMsg = "One of the parameters to the call is not correct."
        Case "80041003"
            strMsg = "Write Protect error"
        Case "80041062"
            strMsg = "Operation failed because the client did not have the necessary security privilege."
        Case "80045001"
            strMsg = "Authentication failure."
        Case "80045002"
            strMsg = "Password not registered."
        Case Else
            strMsg = "Unknown error."
    End Select
    GetErrMsg = strMsg
End Function
TOSHIBA BIOS WMI Interface Guide

Chapter 3 Using the TOSHIBA BIOS WMI Interface

The scripts explained

In this section, we will explain the key points of the scripts and how the WMI Methods and Queries are used. This part of the main script:

```vbs
'create Object to open the procedure file
strFileName = "procedures.vbs"
Set objFSO = CreateObject("Scripting.FileSystemObject")
Set objFile = objFSO.OpenTextFile(strFileName, 1) '1 - for reading
Execute objFile.ReadAll()
```

will load the "procedures.vbs" file containing the additional functions into memory so that they can be called upon. This is a Visual Basic syntax for dynamically loading libraries.

The main script calls the `IsSupervisorPasswordRegistered` function to check if the supervisor password is set.

```vbs
strReturn = IsSupervisorPasswordRegistered(objWMIService)
```

This calls the function in the "procedures.vbs" and the function will read whole password structure into a variable. By using the `GetPasswordStatus` method, the SupervisorPassword is filtered and its status is read.

```vbs
Set colItems = objWMIService.ExecQuery("Select * from Password where InstanceName='ACPI\PNP0C14\0_0'")
For Each objItem in colItems
'execute the method and obtain the return status
objItem.GetPasswordStatus "SupervisorPassword;", strReturn
'return value contains two elements, each seperated by comma.
  e.g: "SupervisorPassword,Registered"
```

Now when the supervisor password is set on the remote computer, the main script needs to authenticate to be able to perform any changes to the remote BIOS.

The script calls `SetConfigurationMode` in the "procedures.vbs" file. The Parameters needed are Start, + the encoded Supervisor Password + ";".

The encoded password can be generated by accessing the following website:

[https://www.biospw.com/tsb/encoder/](https://www.biospw.com/tsb/encoder/)

```vbs
'authenticate with Supervisor privilege
strParameter = "Start," + strSupervisorPassword + ";"
strReturn = SetConfigurationMode(objWMIService, strParameter)
```

The `SetConfigurationMode` function gets a handle on the `ModeControl` structure and uses the method `SetConfigurationMode` to "Start" the authentication.

```vbs
Set colItems = objWMIService.ExecQuery("Select * from ModeControl where InstanceName='ACPI\PNP0C14\0_0'")
For Each objItem in colItems
'execute the method and obtain the return status
objItem.SetConfigurationMode strParameter, strReturn
```
Now the main script is authenticated and can perform changes to the remote BIOS. Get a handle on the `BiosSettings` structure and use the `SetBiosSettings` Method to change the BIOS.

```vbscript
Set colItems = objWMIService.ExecQuery("Select * from BiosSetting where InstanceName='ACPI\PNP0C14\0_0'")

' set single Bios setting
For Each objItem in colItems
    'execute the method and obtain the return status
    objItem.SetBiosSetting strInParamValue, strReturn
Next

When exiting the script - do not forget to end the authentication of the remote supervisor password:

' deauthenticate from supervisor mode
strParameter = "End," + strSupervisorPassword + ";"
strReturn = SetConfigurationMode(objWMIService, strParameter)
```

### Return values

You will receive one of the following return values after calling the WMI methods. Zero is returned on a successful operation. Other values are returned when an error occurs. The following return values are used by the Toshiba WMI interface. Please see the Microsoft Developer Network (MSDN) for other return values.

#### Table 9  Method Return Value

<table>
<thead>
<tr>
<th>Return Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 (0x0) WBEM_S_NO_ERROR</td>
<td>The operation was successful.</td>
</tr>
<tr>
<td>2147749900 (0x8004100C) WBEM_E_NOT_SUPPORTED</td>
<td>Feature or operation is not supported.</td>
</tr>
<tr>
<td>2147749896 (0x80041008) WBEM_E_INVALID_PARAMETER</td>
<td>One of the parameters to the call is not correct The system supports the subfunction but the subfunction is called with the invalid input arguments.</td>
</tr>
<tr>
<td>2147749891 (0x80041003) WBEM_E_ACCESS_DENIED</td>
<td>Write Protect error Examples: To enable CMP (Core Multi Processing) while TxT (Trusted eXecution Technology) is enabled. To change VT (Virtualization Technology) setting while TxT is enabled To enable TxT while either VT or CMP is disabled To change Boot Menu mode while it is restricted by the policy</td>
</tr>
<tr>
<td>2147749986 (0x80041062) WBEM_E_PRIVILEGE_NOT_HELD</td>
<td>Operation failed because the client did not have the necessary security privilege. Examples: To change BIOS settings if the supervisor password is registered but not provided by ConfigurationMode class</td>
</tr>
<tr>
<td>2147766273 (0x80045001) (Toshiba original)</td>
<td>Authentication failure Examples: An incorrect password is provided with ConfigurationMode class. An incorrect password is provided as the old password to change a password</td>
</tr>
<tr>
<td>2147766274 (0x80045002) (Toshiba original)</td>
<td>Password not registered Examples: ConfigurationMode is called but Supervisor password is not registered.</td>
</tr>
</tbody>
</table>
Other methods

Load Default Bios Settings

Most BIOS settings have a default value. There is a method to reset the BIOS settings with a predefined default value back to original state. This can be done with the method: LoadDefaultBiosSettings.

Excerpts out of a VB Script:

```
'connect to WMI
Set objWMIService = GetObject("winmgmts:\" & strComputer & "\root\WMI")

'exectues a WQL query
Set colItems = objWMIService.ExecQuery("Select * from BiosSetting where InstanceName='ACPI\PNP0C14\0_0'")

For Each objItem in colItems
    'execute the method and obtain the return status
    objItem.LoadDefaultBiosSettings "Execute;", strReturn
Next
```

System BIOS Settings

There are two fields in the System BIOS that can be set to store customer asset tag information.

Table 10 Asset Tag BIOS item

<table>
<thead>
<tr>
<th>SMBIOS item</th>
<th>Item name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type2</td>
<td>Asset Tag</td>
</tr>
<tr>
<td></td>
<td>BoardAssetTag</td>
</tr>
<tr>
<td>Type3</td>
<td>Asset Tag Number</td>
</tr>
<tr>
<td></td>
<td>ChassisAssetTagNumber</td>
</tr>
</tbody>
</table>

This script shows how to set any of these two strings in the SMBIOS.

```
'Sample VBScript: Set a single smbios string on the local computer. Use this script if you have no supervisor password set.
'S
    command line: cscript.exe SetSmbiosConfig.vbs [string] [value]
    argument 1 : Smbios item name
    2 : the string value you want to change

'declare application name
Dim strAppName
strAppName = "SetSmbiosConfig"

On Error Resume Next
Dim objWMIService, objItem, colItems, strComputer, strInParamValue, strReturn, strItem, strStatus, strFileName, strParameter

''check input parameters
If WScript.Arguments.Count <> 2 Then
    WScript.Echo "Usage: cscript.exe SetSmbiosConfig.vbs [string] [value]"
```
WScript.Quit
End If

'check write permissions
If WScript.Arguments(0) <> "BoardAssetTag" And
WScript.Arguments(0) <> "ChassisAssetTagNumber" Then
    WScript.Echo "You do not have permission to modify smbios strings except BoardAssetTag and ChassisAssetTagNumber."
    WScript.Quit
End If

'define variables
strInParamValue = WScript.Arguments(0) + "," + WScript.Arguments(1) + ";"
strComputer = "." 'replace your computer name or leave "." as default value

'connect to WMI
Set objWMIService = GetObject("winmgmts:\" & strComputer & "\root\WMI")
If Err.Number <> 0 Then
    WScript.Echo "Unable to connect to WMI service: " & Hex(Err.Number) & "."
    WScript.Quit
End If

'executes a WQL query
Set colItems = objWMIService.ExecQuery("Select * from SmbiosString where InstanceName='ACPI\PNP0C14\0_0'")

' set single smbios string
For Each objItem in colItems
    'execute the method and obtain the return status
    objItem.SetSmbiosString strInParamValue, strReturn
Next

WScript.Echo strAppName & ": " & GetErrMsg(Hex(strReturn))
WScript.Quit

'convert an error code to a string
Function GetErrMsg(err)
    Dim strMsg
    Select Case err
        Case "0"
            strMsg = "The operation was successful."
        Case "8004100C"
            strMsg = "Feature or operation is not supported."
        Case "80041008"
            strMsg = "One of the parameters to the call is not correct."
        Case "80041003"
            strMsg = "Write Protect error"
        Case "80041062"
    End Select
    GetErrMsg = strMsg
End Function
strMsg = "Operation failed because the client did not have the necessary security privilege."
Case "80045001"
    strMsg = "Authentication failure."
Case "80045002"
    strMsg = "Password not registered."
Case Else
    strMsg = "error code " + err
End Select
GetErrMsg = strMsg
End Function
All the BIOS Settings are listed in the following table. The first two columns show the names of the BIOS setting and on which page they appear in the BIOS Setup Screen. The BIOS Setup Screen can be accessed by pressing **F2** during boot time.

Note that not all settings may be available. Many of the settings are dependent on the available hardware. To find out which settings are available please use the QueryBiosSettings query as described in Chapter 3.

Each of BIOS settings have their own WMI interface name, which is listed in the third column. The fourth column shows the access attribute of each item. The “Acceptable values” column lists all values that each item can take. It is recommended to use the QueryBiosSettingsParameter query to list each acceptable value.

The last column shows if a BIOS setting has a default value. The settings marked with Read Only (RO) cannot be set at all. The settings marked with Write Only (WO) can only be written. The settings marked with Y (Yes) has a default value whereas the settings with N (No) will keep the current setting even load default is executed. You can change the BIOS Settings to default using the LoadDefaultBiosSettings method as described in chapter 3.

Table 11  BIOS Settings

<table>
<thead>
<tr>
<th>Page in SETUP</th>
<th>Item name in SETUP</th>
<th>WMI Item name</th>
<th>Attr.</th>
<th>Acceptable values</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main</td>
<td>System BIOS Version</td>
<td>SystemBiosVersion</td>
<td>RO</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>EC Version</td>
<td>ECVersion</td>
<td>RO</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>AMT Setup Prompt</td>
<td>AMTSetupPrompt</td>
<td>RW</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>Language</td>
<td>Language</td>
<td>RW</td>
<td>&quot;English&quot;, &quot;French&quot;</td>
<td>N</td>
</tr>
<tr>
<td>Security</td>
<td>Secure Boot</td>
<td>SecureBoot</td>
<td>RW</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td>TPM</td>
<td>TPM (*1)</td>
<td>RW</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td>Clear TPM Owner</td>
<td>ClearTPMOwner</td>
<td>WO</td>
<td>&quot;Execute&quot;</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Hide TPM</td>
<td>HideTPM</td>
<td>RW</td>
<td>&quot;Yes&quot;, &quot;No&quot;</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td>Boot Menu</td>
<td>BootMenu</td>
<td>RW</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Y</td>
</tr>
<tr>
<td>Power Management</td>
<td>Wake-up on LAN</td>
<td>WakeUpOnLAN</td>
<td>RW</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>Wake-up on LAN on Battery</td>
<td>WakeUpOnLANOnBattery</td>
<td>RW</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>Wake on Keyboard</td>
<td>WakeOnKeyboard</td>
<td>RW</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>Critical Battery Wake-up</td>
<td>CriticalBatteryWakeUp</td>
<td>RW</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>Panel Open - Power On</td>
<td>PanelOpenPowerOn</td>
<td>RW</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>Power on by AC</td>
<td>PowerOnByAc</td>
<td>RW</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>Dynamic CPU Frequency Mode</td>
<td>DynamicCPUFrequencyMode</td>
<td>RW</td>
<td>&quot;DynamicallySwitchable&quot;, &quot;AlwaysHigh&quot;, &quot;AlwaysLow&quot;</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>Core Multi-Processing</td>
<td>CoreMultiProcessing</td>
<td>RW</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>Intel Turbo Boost Technology</td>
<td>IntelTurboBoostsTechnology</td>
<td>RW</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Y</td>
</tr>
<tr>
<td>Feature</td>
<td>Setting</td>
<td>Read/Write</td>
<td>Options</td>
<td>Default</td>
<td></td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>--------------------------------------------</td>
<td>------------</td>
<td>----------------------------------------------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td>Intel Display Power Management</td>
<td>IntelDisplayPowerManagement</td>
<td>RW</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>eSATA</td>
<td>eSATA</td>
<td>RW</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>SATA Interface setting</td>
<td>SATAInterfaceSetting</td>
<td>RW</td>
<td>&quot;Performance&quot;, &quot;BatteryLife&quot;</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Intel(R) Rapid Start Technology</td>
<td>IntelRapidStartTechnology</td>
<td>RW</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Rapid Start Entry after</td>
<td>RapidStartEntryAfter</td>
<td>RW</td>
<td>&quot;Immediately&quot;, &quot;10minutes&quot;, &quot;2hours&quot;, &quot;5hours&quot;, &quot;24hours&quot;</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Internal USB3.0 Controller</td>
<td>InternalUSB30Controller</td>
<td>RW</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Keyboard Backlight Control Mode</td>
<td>KeyboardBacklightControlMode</td>
<td>RW</td>
<td>&quot;TIMER&quot;, &quot;ON&quot;, &quot;OFF&quot;</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Backlight Lighting Time</td>
<td>BacklightLightingTime</td>
<td>RW</td>
<td>&quot;00&quot; - &quot;60&quot;</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Battery Save Mode</td>
<td>BatterySaveMode</td>
<td>RW</td>
<td>&quot;UserSetting&quot;, &quot;LowPower&quot;, &quot;FullPower&quot;</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Processing Speed</td>
<td>ProcessingSpeed</td>
<td>RW</td>
<td>&quot;Low&quot;, &quot;High&quot;</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>CPU Sleep Mode</td>
<td>CPUSleepMode</td>
<td>RW</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>LCD Brightness</td>
<td>LCDBrightness</td>
<td>RW</td>
<td>&quot;Bright, SemiBright&quot;, &quot;SuperBright&quot;</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Cooling Method</td>
<td>CoolingMethod</td>
<td>RW</td>
<td>&quot;CoolingOptimized&quot;, &quot;MaximumPerformance&quot;, &quot;BatteryOptimized&quot;</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>PCI Express Link ASPM</td>
<td>PCIExpressLinkASPM</td>
<td>RW</td>
<td>&quot;Disable&quot;, &quot;Enable&quot;, &quot;Auto&quot;</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Power Management (BIOS Power Management)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Battery Save Mode</td>
<td>BatterySaveMode</td>
<td>RW</td>
<td>&quot;UserSetting&quot;, &quot;LowPower&quot;, &quot;FullPower&quot;</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Processing Speed</td>
<td>ProcessingSpeed</td>
<td>RW</td>
<td>&quot;Low&quot;, &quot;High&quot;</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>CPU Sleep Mode</td>
<td>CPUSleepMode</td>
<td>RW</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>LCD Brightness</td>
<td>LCDBrightness</td>
<td>RW</td>
<td>&quot;Bright, SemiBright&quot;, &quot;SuperBright&quot;</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Cooling Method</td>
<td>CoolingMethod</td>
<td>RW</td>
<td>&quot;CoolingOptimized&quot;, &quot;MaximumPerformance&quot;, &quot;BatteryOptimized&quot;</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Execute-Disable Bit Capability</td>
<td>ExecuteDisableBitCapability</td>
<td>RW</td>
<td>&quot;NotAvailable&quot;, &quot;Available&quot;</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Virtualization Technology</td>
<td>VirtualizationTechnology (*)</td>
<td>RW</td>
<td>&quot;Disable&quot;, &quot;VT-xAndVT-d&quot;, &quot;VT-xOnly&quot;, &quot;VT-dOnly&quot;</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>Trusted Execution Technology</td>
<td>TrustedExecutionTechnology (*)</td>
<td>RW</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>Intel(R) AT</td>
<td>IntelAT</td>
<td>RW</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Intel(R) AT Suspend</td>
<td>IntelATSuspend</td>
<td>RW</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Beep Sound</td>
<td>BeepSound</td>
<td>RW</td>
<td>&quot;OFF&quot;, &quot;Low&quot;, &quot;Medium&quot;, &quot;High&quot;</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Sleep and Charge</td>
<td>SleepAndCharge</td>
<td>RW</td>
<td>&quot;Disable&quot;, &quot;AutoMode&quot;, &quot;20AMode&quot;</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>System ON CDP Charge Mode</td>
<td>SystemOnCDPChargeMode</td>
<td>RW</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>USB Power in Sleep Mode</td>
<td>USBPowerInSleepMode</td>
<td>RW</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Sleep and Music</td>
<td>USBSleepAndMusic</td>
<td>RW</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>USB Legacy Emulation</td>
<td>USBLegacyEmulation</td>
<td>RW</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>USB Memory BIOS Support Type</td>
<td>USBBiosMemoryBIOSSupportType</td>
<td>RW</td>
<td>&quot;HDD&quot;, &quot;FDD&quot;</td>
<td>Y</td>
<td></td>
</tr>
</tbody>
</table>
**Chapter 4 BIOS Settings**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Setting</th>
<th>Options</th>
<th>Default Value</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change Boot Order</td>
<td>ChangeBootOrder</td>
<td>&quot;HDD/SSD&quot;, &quot;USBMemory&quot;, &quot;eSATAHDD&quot;, &quot;USBODD&quot;, &quot;FDD&quot;, &quot;LAN&quot;</td>
<td>Y</td>
<td>(!) This item cannot be changed if TrustedExecutionTechnology is enabled.</td>
</tr>
<tr>
<td>Built-in LAN</td>
<td>BuiltInLAN</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Wireless LAN</td>
<td>WirelessLAN</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Auto Wireless LAN RF Switching</td>
<td>AutoWirelessLANRFSwitching</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Wireless WAN</td>
<td>WirelessWAN</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>WIMAX</td>
<td>WiMAX</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Bluetooth</td>
<td>Bluetooth</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Internal Pointing Device</td>
<td>InternalPointingDevice</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Web Camera</td>
<td>WebCamera</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>SD Host Controller</td>
<td>SDHostController</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Wireless WAN</td>
<td>WirelessWAN</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>WiFi</td>
<td>WiMAX</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Bluetooth</td>
<td>Bluetooth</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Memory Performance Mode</td>
<td>MemoryPerformanceMode</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>SATA Controller Mode</td>
<td>SATAControllerMode</td>
<td>&quot;IDEMode&quot;, &quot;OSAHCI Mode&quot;, &quot;BIOSAHCI Mode&quot;</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>Boot Mode</td>
<td>BootMode</td>
<td>&quot;CSMBoot&quot;, &quot;UEFIBoot&quot;</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>Power On Display</td>
<td>PowerOnDisplay</td>
<td>&quot;Auto-Selected&quot;, &quot;SystemLCDOnly&quot;</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Boot Up NumLock Status</td>
<td>BootUpNumlockStatus</td>
<td>&quot;ON&quot;, &quot;OFF&quot;</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Function Button</td>
<td>FunctionButton</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Function Button Beep</td>
<td>FunctionButtonBeep</td>
<td>&quot;OFF&quot;, &quot;Low&quot;, &quot;Medium&quot;, &quot;High&quot;</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>HDMI-CEC</td>
<td>HDMI-CEC</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Remote Power On / Off</td>
<td>RemotePowerOnOff</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Function Keys mode</td>
<td>FunctionKeysMode</td>
<td>&quot;StandardF1F12mode&quot;, &quot;SpecialFunctionMode&quot;</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>In TPM Control Utility</td>
<td>ShowTPMConfirmationMessage</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>Toshiba Supervisor Registration Utility</td>
<td>ShowHDDPasswordMenu</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>Toshiba Supervisor Registration Utility</td>
<td>OwnerString</td>
<td>-</td>
<td>N</td>
<td></td>
</tr>
</tbody>
</table>

(*1) This item cannot be changed if TrustedExecutionTechnology is enabled.

(*2) To enable this item, it is required to set TPM to Enabled and VirtualizationTechnology to EnabledVT-xAndVT-d.
(*3) This item takes several values as arguments.

(*4) This setting cannot be accessed through the BIOS Setup Utility. It can only be accessed from the TPM Control Utility or through the Toshiba WMI Interface.

(*5) Toshiba Supervisor Registration Utility, which is a DOS based utility. There is no access to these settings through the BIOS Setup Utility. These settings can only be accessed through the Toshiba Supervisor Registration Utility or the Toshiba WMI Interface.
There are special features included in the BIOS that are only active when a supervisor password is set. This includes access to enable/disable certain built-in components and to restrict specific boot devices.

The Toshiba Supervisor Registration Utility, is a standalone utility for modifying the settings intended for the supervisor. The names of the settings in the first column are the names how they are used in the Supervisor Registration Utility. The second column contains the names of the settings accessed through the WMI interface.

Table 12  Settings for the supervisor

<table>
<thead>
<tr>
<th>Name in Toshiba Supervisor Registration Utility</th>
<th>WMI Item name</th>
<th>Acceptable values</th>
<th>Description of the setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>RegistPswd</td>
<td>RegisterPassword</td>
<td>“Enable”, “Disable”</td>
<td>Allow user to register user password</td>
</tr>
<tr>
<td>DeletePswd</td>
<td>DeletePassword</td>
<td>“Enable”, “Disable”</td>
<td>Allow user to delete user password</td>
</tr>
<tr>
<td>ChangePswd</td>
<td>ChangePassword</td>
<td>“Enable”, “Disable”</td>
<td>Allow user to change user password</td>
</tr>
<tr>
<td>NoLockPswd</td>
<td>NeverLockPassword</td>
<td>“Enable”, “Disable”</td>
<td>Does not lock user password even if user password verification exceeds max retry counts</td>
</tr>
<tr>
<td>MaxChekTry</td>
<td>MaxTryCount</td>
<td>“1”-“15”, “Unlimited”</td>
<td>Max retry count to verify user PW (1-15 or Unlimited ). The default value is “3”.</td>
</tr>
<tr>
<td>MinPswdLen</td>
<td>MinimumPasswordLength</td>
<td>“1”-“15”</td>
<td>Minimum length of user password (1-15). The default value is “1”.</td>
</tr>
<tr>
<td>BiosSetup</td>
<td>BIOSSetup</td>
<td>“Enable”, “Disable”</td>
<td>Allow to edit BIOS SETUP (SYSTEM SETUP)</td>
</tr>
<tr>
<td>BiosUpdate</td>
<td>BIOSUpdate</td>
<td>“Enable”, “Disable”</td>
<td>Allow user to update the BIOS</td>
</tr>
<tr>
<td>NotViewMode</td>
<td>ViewMode</td>
<td>“Enable”, “Disable”</td>
<td>“Enable”: User can only read but cannot write BIOS settings</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>NotViewMode(Enabled) = ViewMode(Disabled)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>NotViewMode(Disabled) = ViewMode(Enabled)</td>
</tr>
<tr>
<td>RegDelHDDpw</td>
<td>RegisterHDDPassword</td>
<td>“Enable”, “Disable”</td>
<td>Allow user to register a HDD password (user or master)</td>
</tr>
<tr>
<td>ChangeHDDpw</td>
<td>ChangeHDDPassword</td>
<td>“Enable”, “Disable”</td>
<td>Allow user to change the HDD password</td>
</tr>
<tr>
<td>ActivateTPM</td>
<td>ActivateTPM</td>
<td>“Enable”, “Disable”</td>
<td>Allow to configure “TPM Enable/Disable” in BIOS SETUP</td>
</tr>
<tr>
<td>OwnerClrTPM</td>
<td>OwnerClearTPM</td>
<td>“Enable”, “Disable”</td>
<td>Allow to configure “Clear TPM Owner” in BIOS SETUP</td>
</tr>
<tr>
<td>BTcert</td>
<td>BluetoothAuthentication</td>
<td>“Enable”, “Disable”</td>
<td>Enable/disable BlueTooth authentication</td>
</tr>
<tr>
<td>FPcert</td>
<td>FingerprintAuthentication</td>
<td>“Enable”, “Disable”</td>
<td>Enable/disable FingerPrint authentication</td>
</tr>
<tr>
<td>IoCOM</td>
<td>DeviceSerialPort</td>
<td>“Enable”, “Disable”</td>
<td>Enable/disable Serial Port (RS-232C serial port)</td>
</tr>
<tr>
<td>IoODD</td>
<td>DeviceODD</td>
<td>“Enable”, “Disable”</td>
<td>Enable/disable Optical Disc Drive (internal CD-ROM drive, CD/DVD/HD-DVD/BD multi-drive)</td>
</tr>
<tr>
<td>Name in Toshiba Supervisor Registration Utility</td>
<td>WMI Item name</td>
<td>Acceptable values</td>
<td>Description of the setting</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>--------------------</td>
<td>-------------------</td>
<td>-----------------------------------------------------------------</td>
</tr>
<tr>
<td>IoBluetooth</td>
<td>DeviceBluetooth</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Enable/disable Bluetooth (except for SD/USB Bluetooth)</td>
</tr>
<tr>
<td>IoMODEM</td>
<td>DeviceModem</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Enable/disable Internal Modem</td>
</tr>
<tr>
<td>IoUSB</td>
<td>DeviceUSB</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Enable/disable USB Connector</td>
</tr>
<tr>
<td>IoPCCard</td>
<td>DevicePCCard</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Enable/disable PC Card Slot (disabling this item, boot from a PC Card ATA is also disabled)</td>
</tr>
<tr>
<td>IoSD</td>
<td>DeviceSDCard</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Enable/disable SD Card Slot (disabling this item, boot from SD memory card is also disabled)</td>
</tr>
<tr>
<td>IoExpCard</td>
<td>DeviceExpresscard</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Enable/disable xpressCard Slot</td>
</tr>
<tr>
<td>IoWiredLAN</td>
<td>DeviceWiredLAN</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Enable/disable Internal Wired LAN</td>
</tr>
<tr>
<td>IoWlessLAN</td>
<td>DeviceWirelessLAN</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Enable/disable Internal Wireless LAN</td>
</tr>
<tr>
<td>IoWlessWAN</td>
<td>DeviceWirelessWAN</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Enable/disable Internal Wireless WAN</td>
</tr>
<tr>
<td>IoMediaSlot</td>
<td>DeviceMediaSlot</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Enable/disable Internal Media Slot</td>
</tr>
<tr>
<td>IoESATA</td>
<td>DeviceESATA</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Enable/disable eSATA connector (or eSATA portion of an eSATA+USB connector)</td>
</tr>
<tr>
<td>IoWebcam</td>
<td>DeviceWebcam</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Enable/disable Internal Webcam</td>
</tr>
<tr>
<td>EnSmartCard</td>
<td>DeviceSmartCard</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Enable/disable SmartCard device</td>
</tr>
<tr>
<td>Boot1stHDD</td>
<td>Boot1stHDD</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Enable/disable Boot from 1st Hard Disk Drive</td>
</tr>
<tr>
<td>BootODD</td>
<td>BootODD</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Enable/disable Boot from Optical Disc Drive</td>
</tr>
<tr>
<td>BootFDD</td>
<td>BootFDD</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Enable/disable Boot from Floppy Disk Drive</td>
</tr>
<tr>
<td>BootLAN</td>
<td>BootLAN</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Enable/disable Boot from LAN</td>
</tr>
<tr>
<td>BootUSB</td>
<td>BootUSB</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Enable/disable Boot from USB Memory (USB flash drive and USB Hard Disk Drive)</td>
</tr>
<tr>
<td>BootESATA</td>
<td>BootESATA</td>
<td>&quot;Enable&quot;, &quot;Disable&quot;</td>
<td>Enable/disable Boot from eSATA device</td>
</tr>
</tbody>
</table>

**Note:** The default value of the setting is undelined.

**Note:** If NeverLockPassword is disabled and the user enters the user password incorrectly too many times the computer locks. The computer can only then be unlocked by entering the Supervisor Password.
Troubleshooting

Should you encounter any remote access problems, please try the following solutions.

Checking DCOM permissions

1. Open Component Service by opening Start -> Run and type in Dcomcnfg.
2. Expand Component Service -> Computers -> My computer.
3. Go to the properties of My Computer.
4. Select the COM Security tab.
5. Click Edit Limits under Access Permissions, and ensure Everyone user group has Local Access and Remote Access permission.
6. Click Edit Limit for the launch and activation permissions, and ensure Everyone user group has Local Activation and Local Launch permission.
7. Highlight DCOM Config node, and right click Windows Management and Instruments, then click Properties and check that the used credentials have remote access rights for all options.

Checking permissions for the used credentials to the WMI namespace

1. Click Start -> Run and type in WMlnmgmt.msc, and then click OK.
2. Right click WMI Control, then click Properties.
4. Select Root and click Security.
5. Ensure Authenticated Users has Execute Methods, Provider Right and Enable Account permission; ensure Administrators has all permissions.

Verify WMI Impersonation Rights

1. Click Start -> Run and type in gedit.msc, and then click OK.
4. Verify that the SERVICE account is specifically granted for Impersonate a client after authentication.

Check Network access sharing and security model

1. Click Start -> Run and type in secpol.msc, and then click OK.
2. Expand Local Policies -> Security Options.
3. Check if Network Access: Sharing security model for local accounts is set to “Classic”.
Check Firewall settings

Please refer to the following webpage for more details:

Some general information about remote access

Please refer to the following webpage for more details:

Some general infos about access issues in combination with UAC

Please refer to the following webpage for more details:

Special note for Windows 8

With Windows 8, the UAC settings have been changed. WMI remote access for local Admins is restricted by default and only Domain Admins have guaranteed access.

For testing purpose or in a Workstation environment, follow the steps below to disable UAC for remote Administrators:

1. Click Start -> Run and type in regedit, and then click OK.
2. Expand registry folder:
   - HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System
3. Create a 32-bit DWord entry named LocalAccountTokenFilterPolicy if not existing.
4. Set Value to 1 to disable UAC for remote Administrators.
Visual Basic script to set a password on a remote computer

Please save this script into a file called procedures.vbs. The following script will load this script into memory and use the function `SetConfigurationMode`.

```vbnet
Function SetConfigurationMode(objWMIService, strInParamValue)
    Dim colItems, objItem

    'executes a WQL query
    Set colItems = objWMIService.ExecQuery("Select * from ModeControl where InstanceName='ACPI\PNP0C14\0_0'")

    For Each objItem in colItems
        'execute the method and obtain the return status
        objItem.SetConfigurationMode strInParamValue, strReturn
    Next

    SetConfigurationMode = strReturn
End Function
```

This script can be used for setting, changing or deleting passwords on a remote computer.

'Sample VBScript: Set/Change/Delete a password on a remote computer. (restrictions: cannot set supervisor password, but can modify or delete)

' command line: cscript.exe SetPasswordRemote.vbs [type] "[scrambled old password]" "[scrambled new password]" "[scrambled SupervisorPassword]" [ComputerName|IPAddress] [UserName] [Password]
' argument 1 : password type - user: use if you want to change BIOS User Password,
' supervisor: use if you want to change BIOS Supervisor Password,
' userHDD: use if you want to change User Only HDD Password,
' master+user: use if you want to change Master + User HDD Password
' Master + User HDD Password requires 4 passwords, the first 2 for Master HDD Password and the another 2 for User HDD Password
' 2 : the scrambled 1st old password with quotes (Master HDD Password if choosing Master + User HDD Password)
' 3 : the scrambled 1st new Password with quotes (Master HDD Password if choosing Master + User HDD Password)
'4(optional) : the scrambled 2nd old password
with quotes (User HDD Password if choosing Master + User
HDD Password)
' 5(optional) : the scrambled 2nd new password
with quotes (User HDD Password if choosing Master + User
HDD Password)
' 6 : scrambled supervisor password with quotes
 7 : the IP, the FQDN, or the Computer name of
the client PC you want to access
' 8 : a username that has Administrator
privileges on the client PC you want to access
' 9 : password for the username that has
Administrator privileges on the client PC you want to
access
'declare application name
Dim strAppName
strAppName = "SetPasswordRemote"
On Error Resume Next
Dim argcount, strType, objWMIService, objItem, colItems,
strComputer, strInParamValue, strReturn, strStatus,
strFileName
Dim strOldPassword1, strNewPassword1, strOldPassword2,
strNewPassword2, strSupervisorPassword, strParameter,
strUserName, strPassword
'create Object to open the procedure file
strFileName = "procedures.vbs"
Set objFSO = CreateObject("Scripting.FileSystemObject")
Set objFile = objFSO.OpenTextFile(strFileName, 1)  '1 -
for reading
If objFile is Nothing Then
  WScript.Echo "You can not run this application without
file " & strFileName
  WScript.Quit
End If
Execute objFile.ReadAll()

'check input parameters
argcount = WScript.Arguments.Count
If argcount <> 7 And argcount <> 9 Then
  ShowUsage()
  WScript.Quit
End If
'define variables
If argcount = 7 Then
  Select Case WScript.Arguments(0)
  Case "user"
    strType = "UserPassword"
  Case "supervisor"
    strType = "SupervisorPassword"
  Case "userHDD"
    strType = "UserOnlyHDDPassword"
  Case Else
    ShowUsage()
    WScript.Quit
  End Select
Chapter A Visual Basic script to set a password on a remote computer

```vbscript
strOldPassword1 = WScript.Arguments(1)
strNewPassword1 = WScript.Arguments(2)
strInParamValue = strType + "," + strOldPassword1 + "," + strNewPassword1 + ";"
strSupervisorPassword = WScript.Arguments(3)
strComputer = WScript.Arguments(4) 'computer name or computer's IP address
strUserName = WScript.Arguments(5)
strPassword = WScript.Arguments(6)
Else 'argcount is 9
Select Case WScript.Arguments(0)
Case "master+user"
    strType = "Master+UserHDDPassword"
Case Else
    ShowUsage()
End Select
strOldPassword1 = WScript.Arguments(1)
strNewPassword1 = WScript.Arguments(2)
strOldPassword2 = WScript.Arguments(3)
strNewPassword2 = WScript.Arguments(4)
strInParamValue = strType + "," + strOldPassword1 + "," + strNewPassword1 + "," + strOldPassword2 + "," + strNewPassword2 + ";"
strSupervisorPassword = WScript.Arguments(5)
strComputer = WScript.Arguments(6) 'computer name or computer's IP address
strUserName = WScript.Arguments(7)
strPassword = WScript.Arguments(8)
End If

wbemImpersonationLevelImpersonate = 3
wbemAuthenticationLevelPktPrivacy = 6

'get the locator object
Set objSWbemLocator = CreateObject("WbemScripting.SWbemLocator")

'get the service object from the remote server
Set objWMIService = objSWbemLocator.ConnectServer(strComputer, "root\WMI", strUserName, strPassword)

If Err.Number <> 0 Then
    WScript.Echo "Unable to connect to " & strComputer & ":" & Hex(Err.Number) & "."
    WScript.Quit
End If

objWMIService.Security_.ImpersonationLevel = wbemImpersonationLevelImpersonate
objWMIService.Security_.AuthenticationLevel = wbemAuthenticationLevelPktPrivacy

If strSupervisorPassword <> "" Then
    'authenticate with Supervisor privilege
```

strParameter = "Start," + strSupervisorPassword + ";"
strReturn = SetConfigurationMode(objWMIService, strParameter)
If strReturn <> 0 Then
  WScript.Echo "Supervisor password authentication failed. Error: " & GetErrMsg(Hex(strReturn))
  WScript.Quit
Else
  WScript.Echo "Supervisor password successfully authenticated."
End If
End If

' executes a WQL query
Set colItems = objWMIService.ExecQuery("Select * from Password where InstanceName='ACPI\PNP0C14\0_0'")

' modify the supervisor password
For Each objItem in colItems
  execute the method and obtain the return status
  objItem.SetPassword strInParamValue, strReturn
Next
WScript.Echo strAppName & ": " & GetErrMsg(Hex(strReturn))

' deauthenticate from supervisor mode
If strType = "SupervisorPassword" And strReturn = 0 Then
  strParameter = "End," + strNewPassword1 + ";"
  ' process by using new supervisor password if success of changing password
Else
  strParameter = "End," + strSupervisorPassword + ";"
End If
End If
If strParameter <> "End,;" Then ' if there is no supervisor password, skip deauthentication process
  strReturn = SetConfigurationMode(objWMIService, strParameter)
  If strReturn <> 0 Then
    WScript.Echo "Supervisor password deauthentication failed. Error: " & GetErrMsg(Hex(strReturn))
    WScript.Quit
  Else
    WScript.Echo "Supervisor password successfully deauthenticated."
  End If
End If
End If

WScript.Quit

' usage help
Sub ShowUsage()
  WScript.Echo "Usage: cscript.exe SetPassword.vbs [type]"
  WScript.Echo "  " [scrambled old password]"  " [scrambled new password]"
  WScript.Echo "  " [scrambled old password]"  " [scrambled new password]"
  WScript.Echo "  " [scrambled SupervisorPassword]"  [ComputerName|IPAddress]
  [UserName] [Password]"
WScript.Echo "       [type]       user
use if you want to change BIOS User Password"
WScript.Echo "                   supervisor
use if you want to change BIOS Supervisor Password"
WScript.Echo "                   userHDD
use if you want to change User Only HDD Password"
WScript.Echo "                   master+user
use if you want to change Master + User HDD Password"
WScript.Echo "                   userHDD
use if you want to change User Only HDD Password"
WScript.Echo "                   master+user
use if you want to change Master + User HDD Password"
WScript.Echo "        Master + User HDD Password
requires 4 passwords, the first 2 for Master HDD Password
and the another 2 for User HDD Password"
End Sub

' convert an error code to a string
Function GetErrMsg(err)
    Dim strMsg
    Select Case err
        Case "0"
            strMsg = "The operation was successful."
        Case "8004100C"
            strMsg = "Feature or operation is not
            supported."
        Case "80041008"
            strMsg = "One of the parameters to the call is
            not correct."
        Case "80041003"
            strMsg = "Write Protect error"
        Case "80041062"
            strMsg = "Operation failed because the client
did not have the necessary security privilege."
        Case "80045001"
            strMsg = "Authentication failure."
        Case "80045002"
            strMsg = "Password not registered."
        Case Else
            strMsg = "error code " + err
    End Select

    GetErrMsg = strMsg
End Function
Sample scripts in PowerShell

Some general notes on PowerShell

By default, Windows rejects to execute downloaded unsigned powershell scripts.

To run below example scripts please do the following:

Open a Command window with Administrator privilege and type in powershell set-executionpolicy remotesigned.

In the CMD window, start Powershell scripts with powershell .\scriptname.ps1


Read all BIOS settings and output to the console

# Below sample script will display all available BIOS settings on the standard output
# Format: Item Name, Accessibility, Current Value
#

# Replace “Computername” with the IP, the FQDN, or the Computer name of the client PC you want to access.
# Use “localhost” or remove this line if you want to access the local computer
$strComputername = “Computername”

# Replace “Username” with a username that has Administrator privileges on the client PC you want to access.
# Use the domain administrator username to access a client PC belonging to an Active Directory domain.
# Leave “Username” blank to get prompted for a username
# To avoid getting prompted for the password, remove the authentication part from the script and execute the script under Administrator or Domain Administrator privileges.
$cred = get-credential “Username”

gwmi –namespace “root\wmi” –class “QueryBiosSettings” –credential $cred –computer $strComputername | ForEach-Object {if($_.CurrentSetting -ne “”){Write-Host $_.CurrentSetting}}

Write a single BIOS item

# Below script will change a single BIOS settings
#

# Replace “Computername” with the IP, the FQDN, or the Computer name of the client PC you want to access.
# Use “localhost” or remove this line if you want to access the local computer
$strComputername = “Computername”
# Replace "Username" with a username that has Administrator privileges on the client PC you want to access.
# Use the domain administrator username to access a client PC belonging to an Active Directory domain.
# Leave "Username" blank to get prompted for a username
# To avoid getting prompted for the password, remove the authentication part from the script and execute the script under Administrator or Domain Administrator privileges.
$cred = get-credential "Username"

# Please input Passwords as scrambled keyboard pass code
# Use the Generator utility to convert plain Passwords into scrambled keyboard pass codes
$SvPW = 'Password'

# Input the Item name. E.g. SleepAndCharge
# Consult the manual or use the "QueryBiosItems" Class to get a list of all available BIOS settings
$Item = ""

# Input the parameter. E.g. AutoMode
# Consult the manual or use the "QueryBiosSettingsParameter" Class to get a list of all possible parameter
# Please note that all values are case sensitive
$value = ""

Write-Host "---"

# Authenticate
$result=""
$mode= gwmi  –namespace "root\wmi" -class "ModeControl" –
credential $cred -computer $strComputername | where
{$_._InstanceName -match "ACPI\pnp0c14\0.0"}
$result = $mode.SetConfigurationMode("Start,$SvPW;").Return
if ($result -eq 0)
{
    Write-Host "Successfully authenticated "

    $result=""
    $read = gwmi  –namespace "root\wmi" -class
    "BiosSetting" -credential $cred -computer $strComputername
    | where {$_._InstanceName -match "ACPI\pnp0c14\0.0"}
    # Filter the line for the parameter
    $currentSetting =
    $read.GetBiosSetting($Item+";").CurrentSetting.Split("","",3)
    [2]+"
    Write-Host "Current Setting =" $currentSetting

    # Set new value
    $result = $read.SetBiosSetting("$Item,$value;").Return
    if ($result -eq 0)
    {
        Write-Host "Successfully applied the new setting"
        $currentSetting =
        $read.GetBiosSetting($Item+";").CurrentSetting.Split("","",3)
        [2]+"
    }
Save current BIOS settings to a file

# Below script is saving the current Bios settings into a file
# After changing settings, the script "ReadSavedBiosSettingsfromFileandWriteToBios.ps1" can be used to write the modified settings back into the Bios
#
# Replace “Computername” with the IP, the FQDN, or the Computer name of the client PC you want to access.
# Use “localhost” or remove this line if you want to access the local computer
$strComputername = “Computername”

# Replace “Username” with a username that has Administrator privileges on the client PC you want to access.
# Use the domain administrator username to access a client PC belonging to an Active Directory domain.
# Leave "Username" blank to get prompted for a username
# To avoid getting prompted for the password, remove the authentication part from the script and execute the script under Administrator or Domain Administrator privileges.
$cred = get-credential “Username”

# Read the raw data
$list = ""
$rawlist = gwmi –namespace ”root\wmi” –class "QueryBiosSettings" –credential $cred –computer $strComputername

foreach ($item in $rawlist)
{
  # Remove empty rows and not supported functions
  Write-Host $item
}
```powershell
if(($item.CurrentSetting -ne "") -and
($item.CurrentSetting -notlike "*NotSupported")
{
    $list += $item.CurrentSetting +"\n"
}

# Replace "CurrentBiosSettings.txt" with your path and filename
Set-Content CurrentBiosSettings.txt $list
Write-Host "Current Bios Settings successfully saved"
```

Reads saved BIOS settings from a file and writes it back to BIOS

# Below script is reading Bios settings from a file and writing them back into the Bios
# Together with the previous script it can be used to realize a backup function but also to modify a bunch of Bios settings in one loop
#
# Each Bios item should be present in one line. You need to follow this structure:
# BiosItemA, settingA1, settingA2, ....
# BiosItemB, settingB1, settingB2, ....
# BiosItem.......
#
# Alternatively, the output of the script "SaveCurrentBiosSettingsToFile.ps1" can be used as a base
#
# Replace “Computername” with the IP, the FQDN, or the Computer name of the client PC you want to access.
# Use “localhost” or remove this line if you want to access the local computer
$strComputername = "Computername"

# Replace “Username” with a username that has Administrator privileges on the client PC you want to access.
# Use the domain administrator username to access a client PC belonging to an Active Directory domain.
# Leave "Username" blank to get prompted for a username
# To avoid getting prompted for the password, remove the authentication part from the script and execute the script under Administrator or Domain Administrator privileges.
$cred = get-credential "Username"

# Replace "CurrentBiosSettings.txt" with your path and filename
$file = "CurrentBiosSettings.txt"

# Please input Passwords as scrambled keyboard pass code
# Use the Generator utility to convert plain Passwords into scrambled keyboard pass codes
$SvPW = 'Password'
```
# Remove the troublemaking empty line from the end of the input file, generated by the Power Shell Set-Content function.
$list = gc $file | where {$_ -ne ""}

Write-Host "-***-
"

# Authenticate
# Notice: backslash must be escaped by "\\"
$result = ""
$mode = gwmi -namespace "root\wmi" -class "ModeControl" -credential $cred -computer $strComputername | where {$_.InstanceName -match "ACPI\pnp0c14\0.0"}
$result = $mode.SetConfigurationMode("Start,$SvPW;").Return

if ($result -eq 0)
{
    Write-Host "Successful Authenticated"

    # Get access to the Bios Setting Class
    $function = gwmi -namespace "root\wmi" -class "BiosSetting" -credential $cred -computer $strComputername | where {$_.InstanceName -match "ACPI\pnp0c14\0.0"}
    $result = ""
    foreach ($item in $list)
    {
        # Split the input line into 3 substrings
        # string[0] = Parameter
        # string[1] = Read/Write indicator
        # string[2] = The parameter value(s) separated by comma
        $para = $item.Split(",",3)[0]+","+$item.Split("",3)[2]+";"

        # Check if setting is writable and restore setting if writable
        if ($item.Split("","",3)[1] -ne "RO")
        {
            $result = $function.SetBiosSetting("$para").Return
            if ($result -eq 0)
            {
                Write-Host "Successfully restored setting for: ", $item.Split("","",3)[0], ": Setting = ", $item.Split("","",3)[2], 
            } else
            {
                Write-Host "Could not write the value, maybe there are dependencies with other settings: ", $item.Split("","",3)[0], ". Error Code: ", $result
            }
        }
    }

    # Deauthentication
    $result = ""
    $result = $mode.SetConfigurationMode("End" +",$SvPW;").Return

if ($result -eq 0) {Write-Host "Successfully deauthenticated"}
else {Write-Host "Deauthentication failed, Error Code= "$result"
}

Write-Host "\nPlease perform a reboot to apply the setting"
}
else {Write-Host "Authentication failed, Error Code= "$result"

Write-Host "\n-***-"

Set or Change BIOS Passwords
#
# Below is a sample script to change or delete the Supervisor or User Password
# Please notice that for security reasons, the Supervisor Password must be set manual or by a separate tool initially (e.g. WinPE based utility) to be able to access the Bios Passwords remotely
#
# Replace “Computername” with the IP, the FQDN, or the Computer name of the client PC you want to access.
# Use “localhost” or remove this line if you want to access the local computer
$strComputername = “Computername”
# Replace “Username” with a username that has Administrator privileges on the client PC you want to access.
# Use the domain administrator username to access a client PC belonging to an Active Directory domain.
# Leave "Username" blank to get prompted for a username
# To avoid getting prompted for the password, remove the authentication part from the script and execute the script under Administrator or Domain Administrator privileges.
$cred = get-credential “Username”
# Define the access level (User or Supervisor)
# - SupervisorPassword
# - UserPassword
# Change below to "UserPassword" for changing / setting / deleting the User Password
$Item = "SupervisorPassword"

# Please input Passwords as scrambled keyboard pass code
# Use the Generator utility to convert plain Passwords into scrambled keyboard pass codes
# Set new Password blank to delete the Password
$oldPW = ‘oldPassword’
$newPW = ‘newPassword’

# Please input below the actual Supervisor Password.
# When changing the Supervisor password, $oldPW and $SvPW are identical.
$SvPW = ‘Password’
Write-Host "-***-`n"

# Authenticate with Supervisor privilege
$result = ""
$mode = gwmi -namespace "root\wmi" -class "ModeControl" -credential $cred -computer $strComputername | where {$_._InstanceName -match "ACPI\pnp0c14\0.0"}
$result = $mode.SetConfigurationMode("Start,$SvPW;").Return

if ($result -eq 0)
{
    Write-Host "Successful Authenticated"

    # Access function
    $function = gwmi -namespace "root\wmi" -class "Password" -credential $cred -computer $strComputername | where {$_._InstanceName -match "ACPI\pnp0c14\0.0"}

    # Set or change the Password
    $result= ""
    $result = $function.SetPassword("$Item,$oldPW,$newPW;").Return
    if ($result -eq 0) {Write-Host "Successfully changed the Password"}
    else {Write-Host "Fail, could not change the Password, Error Code= "$result"}

    $result ="
    # adjust de-authentication PW if access level is Supervisor
    if($item -like "SupervisorPassword") {$SvPW = $newPW}

    # Deauthenticate
    $result = $mode.SetConfigurationMode("End","$SvPW;").Return
    if ($result -eq 0) {Write-Host "Sucessfully deauthenticated"}
    else {
        # Check if $newPW is empty. If so, it means it is removed
        if ($newPW -eq "")
        {
            Write-Host "Password removed"
        }
        Else
        {
            Write-Host "Deauthentication failed, Error Code= "$result"
        }
    }
}

Write-Host `nPlease perform a reboot to apply the setting"

else {Write-Host "Authentication failed, Error Code= "$result"}