Use WMI Explorer* to Program the Ring LED and Button LED

Documentation

Content Type	Install & Setup
Article ID	000023426
Last Reviewed	11/15/2019

Intel® NUC Kits NUC7i[x]BN and NUC6CAY come with Windows Management Instrumentation (WMI) Explorer* in the BIOS. With WMI Explorer, you can query and control the ring LED and button LED from the operating system environment. The WMI Explorer allows you to browse and view WMI namespaces, classes, instances, and properties in a single pane of view.

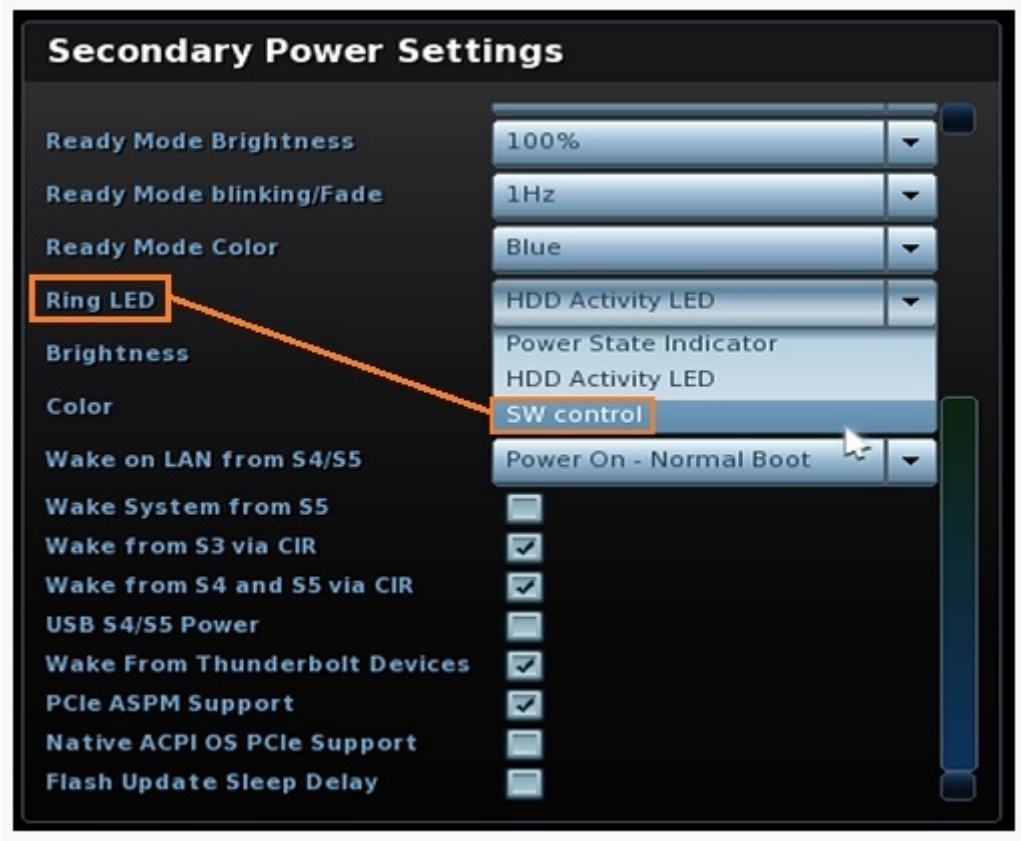
To Use WMI Explorer*

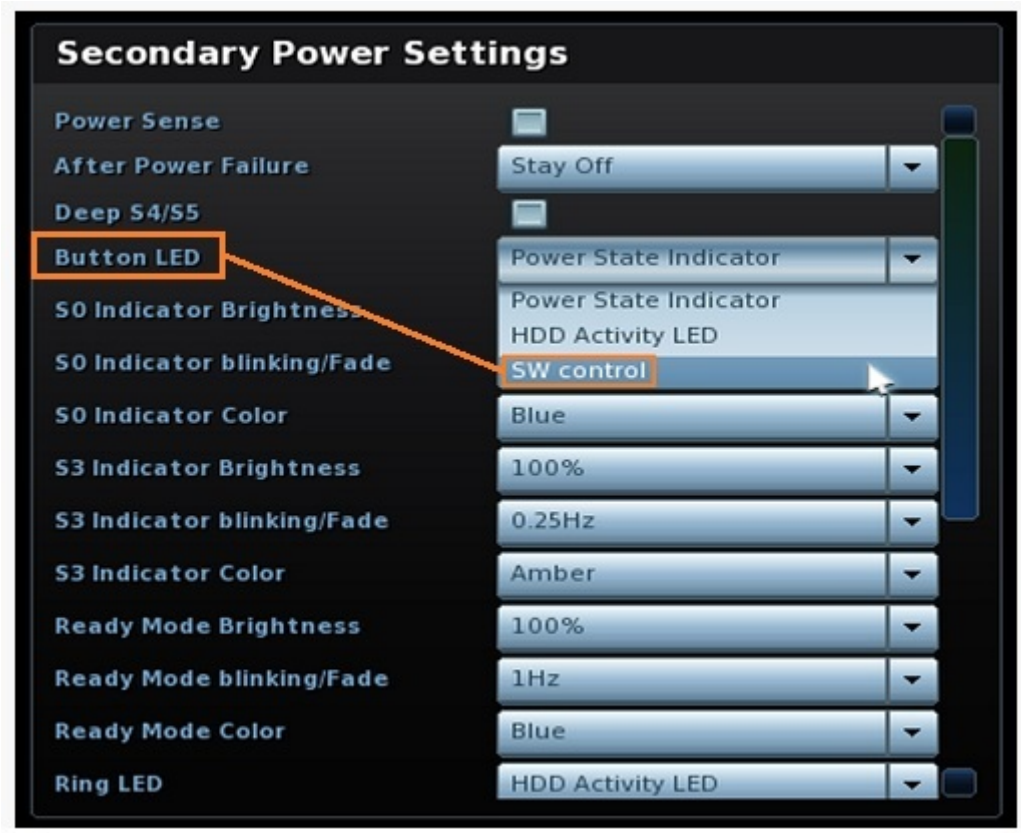
Click or the topic for details: Expand all

Enable LED software control

To enable the LED software control option in the BIOS before programming:

1. Press **F2** during boot to enter BIOS Setup.
2. Go to **Advanced** > **Power**.
3. In the Secondary Power Settings pane, set *Ring LED* and/or *Button LED* to **SW control.**





4. Press **F10** to save and exit BIOS Setup.

MOF to access LED programming tool

We recommend WMI providers implement new WMI classes in Managed Object Format (MOF) files.
We provide the specific MOF with the ASL code for you to access the LED programming tool in the WMI Explorer.

Use this MOF below to access the LED programming tool in the WMI Explorer:

1. GUID: 8C5DA44C-CDC3-46b3-8619-4E26D34390B7
2. _UID: 0
3. Object ID (AA): 65, 65
4. Instance Count: 1
5. Description: Method for get or set the Button or Ring LED state.
6. WMI Object Name: “CISD_WMI”
7. Method ID(1): “GetState”
8. Parameter 1: UINT32(Input)
9. Parameter 2: Package(Array Data)
10. Method ID(2): “SetState”
11. Parameter 1: UINT32(Input)
12. Parameter 2: Package(Array Data)

WMI Method Name and Arguments for LED Control

Get LED Status Function

Control Method		**WMAA**
Arg 0		Instance
Arg 1		Method ID (01h) Get Function
Arg 2 Input Parameter	Byte 0	Function Number 01h - Get S0 Power LED command code
	Byte 1	Reserved
	Byte 2	Reserved
	Byte 3	Reserved

Return Value	Byte 0	Return Code 00h: No Error. Byte 1~3 offer the state. E1h: Error (Function not supported) E2h: Error (Undefined device) E3h: Error (EC doesn't respond) E4h: Error (Invalid Parameter) EFh: Error (Unexpected error) Others: Reserved
	Byte 1	Current LED Brightness state 00h: 0% ~ 64h: 100%
	Byte 2	Using BIOS AY0029 or BN0042 Current LED blinking / fade state 01h: 1Hz 02h: 0.25Hz 03h: Fade 04h: Always on Using BIOS AY0038 or BN0043 and later Current LED blinking / fade state 01h: 1Hz 02h: 0.25Hz 03h: 1Hz fade 04h: Always on 05h: 0.5Hz 06h: 0.25Hz fade 07h: 0.5Hz fade

	Byte 3	Current LED color state Button LED Color: (Get Power LED setting) 00h: Disable 01h: Blue 02h: Amber Ring LED Color: (Get Ring LED setting) 00h: Disable 01h: Cyan 02h: Pink 03h: Yellow 04h: Blue 05h: Red 06h: Green 07h: White

Set LED Function

Control method		**WMAA**
Arg 0		Instance
Arg 1		Method ID (02h) Set LED Function
Arg 2 Input Parameter	Byte 0	Select the LED 01h – Set S0 Power LED command code 02h – Set S0 Ring LED command code
	Byte 1	LED brightness setting 00h: 0% ~ 64h: 100%

	Byte 2	Using BIOS AY0029 or BN0042 Current LED blinking / fade state 01h: 1Hz 02h: 0.25Hz 03h: Fade 04h: Always on Using BIOS AY0038 or BN0043 and later Current LED blinking / fade state 01h: 1Hz 02h: 0.25Hz 03h: 1Hz fade 04h: Always on 05h: 0.5Hz 06h: 0.25Hz fade 07h: 0.5Hz fade
	Byte 3	LED color setting Button LED Color: (Get Power LED setting) 00h: Disable 01h: Blue 02h: Amber Ring LED Color: (Get Ring LED setting) 00h: Disable 01h: Cyan 02h: Pink 03h: Yellow 04h: Blue 05h: Red 06h: Green 07h: White
Return Value	Byte 0	Error Code of Ring LED brightness 00h: No Error E1h: Error (Function not support) E2h: Error (Undefined device) E3h: Error (EC no respond) E4h: Error (Invalid Parameter) EFh: Error (Unexpected error) Others: Reserved

	Byte 1	Error Code of Ring LED blinking/fade 00h: No Error E1h: Error (Function not support) E2h: Error (Undefined device) E3h: Error (EC no respond) E4h: Error (Invalid Parameter) EFh: Error (Unexpected error) Others: Reserved
	Byte 2	Error Code of Ring LED color 00h: No Error E1h: Error (Function not support) E2h: Error (Undefined device) E3h: Error (EC no respond) E4h: Error (Invalid Parameter) EFh: Error (Unexpected error) Others: Reserved
	Byte 3	Reserved

Sample code to invoke the SetState Method

Example of sample code to invoke the SetState Method in the LED WMI interface:

```
public void ChangeRingLED(byte[] data)
{
   object result = new object();
   object[] obj = new object[1];

   obj[0] = BitConverter.ToInt32(data, 0);

   try
   {
      EnumerationOptions options = new EnumerationOptions();
      options.ReturnImmediately = false;

      //specify the WMI Class to call
      ManagementObjectSearcher searcher = new ManagementObjectSearcher("\\root\\WMI", "SELECT * FROM CISD_WMI", options);

      //loop through the objects (there should be only one object, but it might be a good idea to add additional verification)
      foreach (ManagementObject queryObj in searcher.Get())
      {
         result = queryObj.InvokeMethod("SetState", obj);
         queryObj.Dispose();
      }
      EventLog.WriteEntry(sSource, "The Set LED State finished.");

      //if you need to return a result, change the return type of this method and format the result object as needed
   }
   catch (Exception ex)
   {
      // log the error. Reasons for the error may include WMI not running/corrupt on the system, CISD_WMI and/or SetState not installed on the system
      EventLog.WriteEntry(sSource, "The Set LED State failed with error: " + ex.Message.ToString());
   }
}
```

For more on how to write code to call into WMI, see:

- Windows Instrumentation: WMI and ACPI
- Sample C# code for using the latest WMI classes to manage Windows Storage
- Calling a WMI Method
- How to Call a WMI Class Method by Using System.Management

Related Products

This article applies to 13 products.

Show all ❯

Need more help?

Contact support

Give Feedback

Company Information

Our Commitment

Communities

Investor Relations

Contact Us