What are the possible Status values for Win32_ComputerSystem?

devblogs.microsoft.com/oldnewthing/20200818-00

August 18, 2020

Raymond Chen

The documentation for the Win32_ComputerSystem class contains the following documentation for the Status property:

> **Status**
>
> Data type: **string**
>
> Access type: Read-only
>
> Qualifiers: MaxLen (10), DisplayName (“Status”)
>
> Current status of an object. Various operational and nonoperational statuses can be defined. Operational statuses include: OK, Degraded, and Pred Fail, which is an element such as a SMART-enabled hard disk drive that may be functioning properly, but predicts a failure in the near future. Nonoperational statuses include: Error, Starting, Stopping, and Service, which can apply during mirror-resilvering of a disk, reloading a user permissions list, or other administrative work. Not all status work is online, but the managed element is not OK or in one of the other states.
>
> This property is inherited from **CIM_ManagedSystemElement**.
>
> Values include the following:
>
> **OK** (“OK”)
>
> **Error** (“Error”)
>
> **Degraded** (“Degraded”)
>
> **Unknown** (“Unknown”)
>
> **Pred Fail** (“Pred Fail”)
>
> **Starting** (“Starting”)
>
> **Stopping** (“Stopping”)
>
> **Service** (“Service”)
>
> **Stressed** (“Stressed”)
>
> **NonRecover** (“NonRecover”)
>
> **No Contact** (“No Contact”)
>
> **Lost Comm** (“Lost Comm”)

What is all this gibberish trying to say? And what does it tell us about the possible Status values for a Win32_ComputerSystem?

Okay, first the answer, and then the discussion.

The answer is that the Status for a Win32_ComputerSystem is always “OK”. The computer system itself does not report any interesting status for itself, though it does report status for other things, like FrontPanelResetStatus

So why does the Win32_ComputerSystem even have a Status property if it reports nothing interesting?

The clue for that is in the comment that snuck in between the description and the list of status values:

> This property is inherited from **CIM_ManagedSystemElement**.

The Win32_ComputerSystem has a Status property not because it wants to report its own status, but because it is forced to report its own status due to inheritance.

This also explains the rambling description: That description was also inherited from CIM_ ManagedSystemElement. The CIM_ManagedSystemElement class needs to describe all of the possible managed system elements, so it has a large variety of status values available, and the documentation tries to explain what those status values mean as they could be applied to arbitrary derived classes.

Some reformatting and additional clarifying words might help:

> Current status of an object.
>
> Various operational and nonoperational statuses can be defined.
>
> Operational statuses include: OK, Degraded, and Pred Fail. (Pred Fail is reported by an element such as a SMART-enabled hard disk drive that may be functioning properly, but predicts a failure in the near future.)
>
> Nonoperational statuses include: Error, Starting, Stopping, and Service. (Service can be reported during mirror-resilvering of a disk, reloading a user permissions list, or other administrative work. Such work may force the element to go offline temporarily.)
>
> This property is inherited from **CIM_ManagedSystemElement**. Not all status values described here may apply to a derived class.

But really, the description for that property should be marked as non-inherited (even though the property is inherited), and a custom description used instead:

> Current status of an object.
>
> For **Win32_ComputerSystem** the Status is always “OK”.
>
> This property is inherited from **CIM_ManagedSystemElement**.

And that’s what the documentation says as of last week.

Raymond Chen

Follow