Hey, Scripting Guy! Blog

Learn about Windows PowerShell

The 2011 Scripting Games Advanced Event 4: Use PowerShell to Investigate the SvcHost Process

ScriptingGuy1 7 Apr 2011 1:15 AM 8

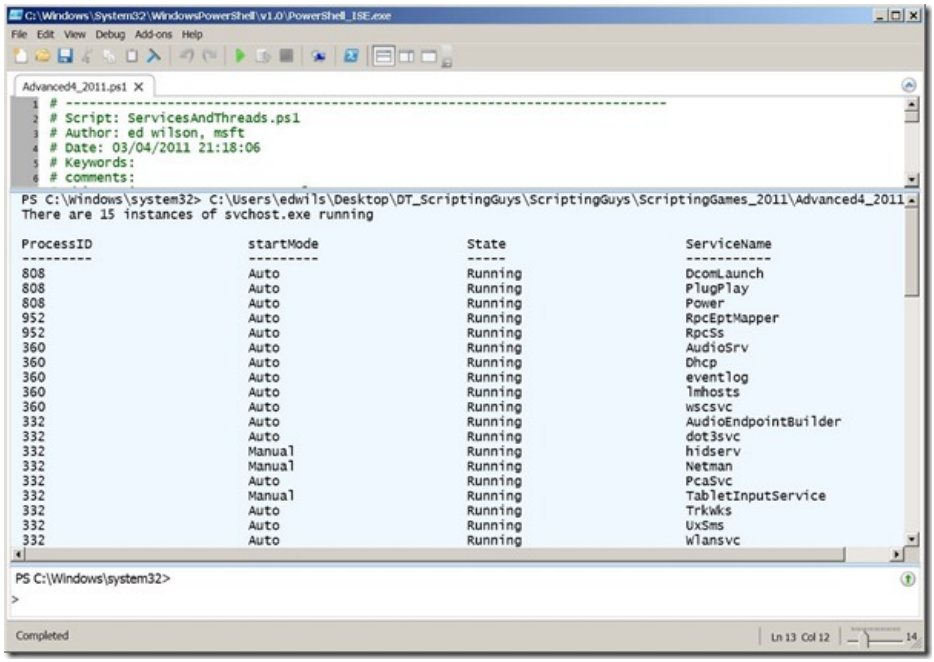
Summary: Advanced Event 4 of the 2011 Scripting Games uses Windows PowerShell to investigate the SvcHost process.

About this event

Division	Advanced
Date of Event	4/7/2011 12:15 AM
Due Date	4/14/2011 12:15 AM

Event scenario

You are the network administrator for a large company with multiple locations around the world. Your performance team has expressed concern with the large amount of memory that is consumed by one particular instance of the SvcHost process. To investigate this process, the team lead has requested that you write a Windows PowerShell script that will list each instance of the SvcHost process, the amount of committed memory, the number of page faults, and the command line that launched each SvcHost process. In addition, the lead requires that the script list each service that is running inside each instance of the SvcHost process. An appropriate output from the script is shown in the following image.



Design points

- Because the performance team is only concerned with a single server, you do not need to provide provisions to run against multiple servers.
- Extra design points for reusable code
- Extra design points for a server that will produce a written report
- Extra design points for a script that returns objects instead of string data or a table.

2011 Scripting Games links

2011 Scripting Games: All Links on One Page

Submit your scripts on PoshCode

Support our Sponsors!

I invite you to follow me on Twitter and Facebook. If you have any questions, send email to me at scripter@microsoft.com, or post your questions on the Official Scripting Guys Forum. Good luck as you compete in this year's Scripting Games. We wish you well.

Ed Wilson, Microsoft Scripting Guy

Comments

7 Apr 2011 10:35 AM

baba

aouch! i was in need of this script! not 4 a game! :(

7 Apr 2011 12:36 PM

Nuno Mota

Another interesting one! :)

However, the event asks us to "list each instance of the SvcHost process, the amount of committed memory, the number of page faults, and the command line that launched each SvcHost process" but the output shown has none of that... What should the output exactly be then?

Thank you!

7 Apr 2011 12:53 PM

Tom

Could you expand on the value you wish to obtain when you ask for committed memory?

IamMred 7 Apr 2011 5:41 PM

@baba the 2011 Scripting Games are more than a game -- they are a learning opportunity. We will have an expert solution to this event in a couple of weeks. In addition, you will be able to see the solutions postedd by the participants in these games in a week.

@nuno you should do exactly what is in the text ... unfortunately, when I took the screen shot some of the columns are cut off (see the scroll bar at the bottom).

@Tom committed memory is the amount of memory that a process reserves when it starts -- it is committmited to that process. A process may reserve 50 MB of memory, and currently only be using 5 MB of memory. If the process never uses the amount of memory it requests then it is being greedy and wasting your system resources. In this case, the process requested TEN Times the amount of memory it needed to do the job. Later, however, it may require additional memory and the commit ensures that the memory is there when it needs it. Some applications provide the capability to tweak the amount of memory the process reserves (typically via the registry). For example, if the process that requests 50 MB never uses more than 20 MB of memory on your machine (because perhaps you do not use all the features of tha application) then it would make sense to tweak the settings for that process and therefore free up 30 MB of memory for your system. Developers often take the view that hardware is cheap, but it does not take too many poorly configured applications to bring even the most robust laptop to its knees when everything wants to start up automatically, and requests huge chunks of committed memory.

7 Apr 2011 9:32 PM

JV

'Committed' is stil ambiguous. BytesReserved and Image BytesReserved.

'Committed' usually only refers to the OS. It can also be used to indicate how many pages are allocated in the page file.

'Committed' here probably indicates how many bytes are requested but has nothing to do with waste. Only actually used bytes will use resources. Yhis number is usally used as a limit. Thart is we do not want to use more than this and don't allocate more. More might mean a memory leak.

WorkingSet is a better metric as it tracks the process. WOrkingSetMax will indicate high demand but is only useful over time.

I think you really want to say that we want to display the BytesReserved or BytesInageReserved...

---

10 Apr 2011 9:38 AM

tom

Can you PLEASE clarify what figures i need to add, or what property i need to pull to get "committed memory". I've been waiting days for a useful response...

as much as i appreciate the explanation, i asked for an answer not an explanation.

IamMred 10 Apr 2011 9:33 PM

@JV yes you are right. Working Set is what I had in mind ... I really did not / do not want to tell you exactly WHICH property to report because it makes the scenario too easy ... I really want you to examine what is being reported, and choose the appropriate property. Compare with task manager if you wish ... look up the class on MSDN ... use get-member to show the class being reported, and then see which property is best to use. This is one reason I "hid" the column in my output.

@Tom you are working with a process named svchost, you will need to use PowerShell to get that process. There are several processes named SVCHOST on your computer, and therefore simply choose one ... any of them will do. Then send the output to a list and look at ALL of the properties to see what kind of information is returned. You can compare your output with the output displayed in Task Manager (right click on your task bar and choose "start task manager"). In task manager, there is view / Select columns. Choose memory - commit size. Note the commit memory displayed in task manager for your process. Then find that amount of memory being used in PowerShell for your process ... Keep in mind that Task manager displays in Kilobytes and PowerShell displays in bytes. You can use 1KB as a constant to "translate bytes into KiloBytes like this: if I have 1024 bytes and I want to convert it to Kilobytes I do this: 1024 / 1KB the result is 1.

---

hassan sayed issa20014 4 Aug 2014 8:57 PM

thanks
