

# Runaway Processes

Michael Morales

(Reprinted From WindowsItPro Magazine)

**Executive Summary:** Runaway processes are a common cause of PC performance problems, but solving the problem requires more than just stopping the process: You have to find what's causing the runaway process. Two free Microsoft tools, Process Explorer and Adplus, can help you troubleshoot runaway-process problems and resolve them faster by revealing which components within a process are consuming the most memory. Learn how to use Process Explorer and Adplus by walking through an example runaway-process troubleshooting scenario.

As a Microsoft escalation engineer, I've fielded many calls about runaway processes—a common cause of PC performance problems. You probably already know what a runaway process is: It's a process that consumes one or more processors, causing your system to become sluggish and sometimes causing other applications to freeze or crash. Often you can find such problems quickly by using Task Manager to view information about running processes. The common way to “solve” a runaway-process problem is to just kill the process, but this is only a temporary fix; the runaway process will likely recur.

To truly fix the problem, you might need to enlist the application manufacturer's tech support. However, before you pick up the phone, there are things you can do to make the support call a lot shorter or perhaps avoid it altogether. I'll tell you about a couple of free tools that Microsoft Support uses to troubleshoot runaway processes and show how you can use them to do the same thing at your site.

## Two Troubleshooting Tools

To troubleshoot processes that consume large amounts of CPU clock cycles, we use two main tools: Process Explorer and Adplus. Each tool has its advantages and best-use scenarios. Process Explorer is available at [live.sysinternals.com](http://live.sysinternals.com). This is the perfect tool to use when you're seeing warning signs that a process will become a runaway and you have either remote or physical access to the system. Adplus is a script file and comes installed with the Debugging Tools for Windows ([www.microsoft.com/whdc/devtools/debugging/default.aspx](http://www.microsoft.com/whdc/devtools/debugging/default.aspx)). This is the tool of choice if you don't know exactly when the process will start to run away and you don't have console access when the problem happens.

## Using Process Explorer

Here's an actual case from the Microsoft Escalation Services files that shows you how to use these tools. We recently resolved an issue where the Windows XP SP2 `wmiprvse.exe` process (a separate host process for WMI providers) was spiking the CPU. The customer had already used Performance Monitor, choosing the %Processor Time counter for the `wmiprvse.exe` process, to identify which process was spiking the CPU.

In this scenario, we knew when the problem would occur, and the customer had physical access to the workstation when the process spike occurred, so our next step was to use Process Explorer to learn more about what components were involved during the spike.

After you've opened Process Explorer, your next step is to configure the location of symbols in Process Explorer. Symbols are used to convert binary information into a readable format. (I discuss

## Runaway Processes

Michael Morales

(Reprinted From WindowsItPro Magazine)

symbols in more detail in “Resolve Memory Leaks Faster,” October 2008, InstantDoc ID 99933.) But before you configure the symbols’ location, make sure that the Debugging Tools for Windows are already installed on your system; you’ll use part of this toolset to make your examination of the suspect process seamless.

To configure the symbols’ location, in Process Explorer click Options, then click Configure Symbols. If you installed the debugging tools in the default location, the Configure Symbols dialog box should display the symbol path.

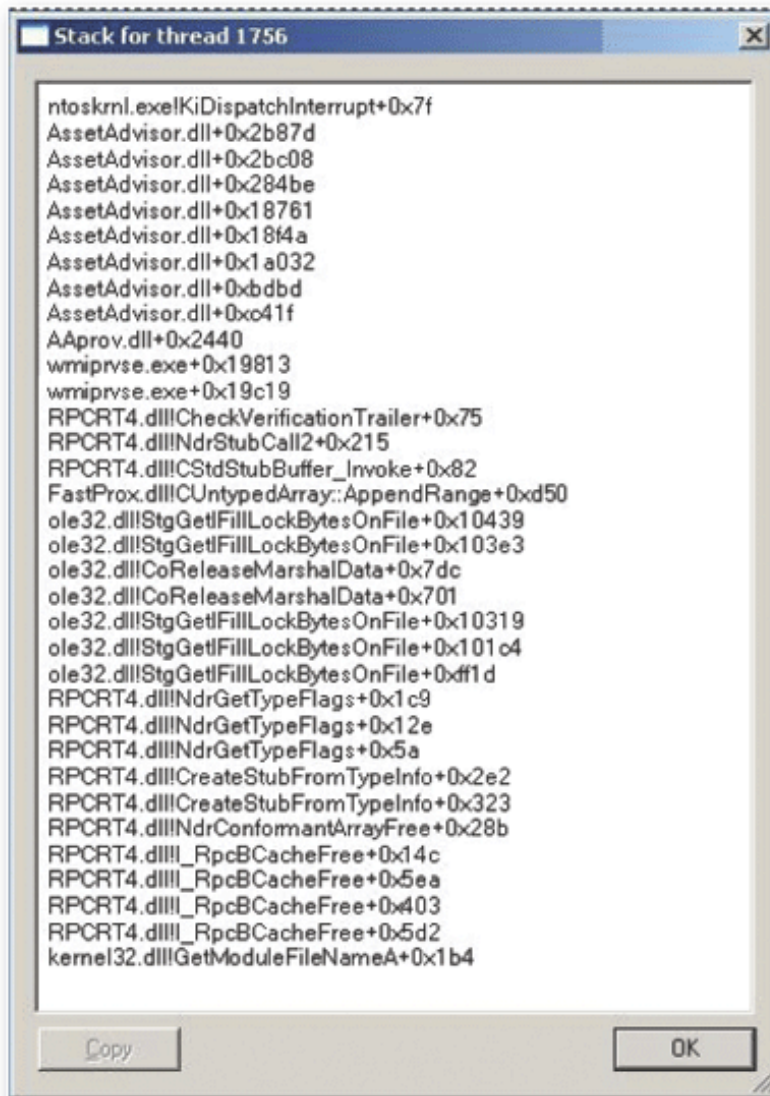
Now you’re ready to dig into the components within the process that are involved in spiking the CPU. From the list of processes displayed in Process Explorer, locate the spiking process (wmiprvse .exe) and double-click it. You’ll see the process’s Properties dialog box, which contains all the threads inside that process with the thread consuming the most CPU resources at the top of the list. Next click the Stack button to get the module and function-call listing for the highest-CPU-consuming thread. You’ll see a screen like the one that Figure 1, shows.

In Figure 1 you can see several modules, and you can use their information to start narrowing down the component (or components) that could have caused the runaway process and possibly solve the problem by researching it online. For example, I did a web search on the string “Assetadvisor.dll wmiprvse cpu” because the CPU spike occurred inside the wmiprvse.exe process and Asset Advisor.dll was a component on the stack and repeated several times. My search found a TechNet article providing a fix for the problem at [support.microsoft.com/kb/937882](http://support.microsoft.com/kb/937882). Thus, when you select components to search, you’ll probably be most successful searching those that are both high on the thread stack and repeated. Identifying what components are involved in a spike will help you in researching the problem on your own or minimizing the length of a support call, if one is needed.

## Runaway Processes

Michael Morales

(Reprinted From WindowsItPro Magazine)



```
Stack for thread 1756
ntoskrnl.exe!KiDispatchInterrupt+0x7f
AssetAdvisor.dll+0x2b87d
AssetAdvisor.dll+0x2bc08
AssetAdvisor.dll+0x284be
AssetAdvisor.dll+0x18761
AssetAdvisor.dll+0x18f4a
AssetAdvisor.dll+0x1a032
AssetAdvisor.dll+0xbdbd
AssetAdvisor.dll+0xc41f
AApprov.dll+0x2440
wmiprvse.exe+0x19813
wmiprvse.exe+0x19c19
RPCRT4.dll!CheckVerificationTrailer+0x75
RPCRT4.dll!NdrStubCall2+0x215
RPCRT4.dll!CStdStubBuffer_::Invoke+0x82
FastProx.dll!CUntypedArray::AppendRange+0xd50
ole32.dll!StgGetFillLockBytesOnFile+0x10439
ole32.dll!StgGetFillLockBytesOnFile+0x103e3
ole32.dll!CoReleaseMarshalData+0x7dc
ole32.dll!CoReleaseMarshalData+0x701
ole32.dll!StgGetFillLockBytesOnFile+0x10319
ole32.dll!StgGetFillLockBytesOnFile+0x101c4
ole32.dll!StgGetFillLockBytesOnFile+0xff1d
RPCRT4.dll!NdrGetTypeFlags+0x1c9
RPCRT4.dll!NdrGetTypeFlags+0x12e
RPCRT4.dll!NdrGetTypeFlags+0x5a
RPCRT4.dll!CreateStubFromTypeInfo+0x2e2
RPCRT4.dll!CreateStubFromTypeInfo+0x323
RPCRT4.dll!NdrConformantArrayFree+0x28b
RPCRT4.dll!_RpcBCacheFree+0x14c
RPCRT4.dll!_RpcBCacheFree+0x5ea
RPCRT4.dll!_RpcBCacheFree+0x403
RPCRT4.dll!_RpcBCacheFree+0x5d2
kernel32.dll!GetModuleFileNameA+0x1b4
```

Figure 1: Module and function-call listing for highest-CPU-consuming thread

Remember that Process Explorer worked in our scenario because of two specific conditions:

- We knew when the process would spike the CPU.
- We had access to the affected system's console.

### Using Adplus

If your runaway-process scenario doesn't match both of those conditions, you'll want to consider using Adplus. Adplus lets you gather the same type of information as Process Explorer, but you don't need

## Runaway Processes

Michael Morales

(Reprinted From WindowsItPro Magazine)

to know when the process will spike the CPU or have physical access to the system during the problem. I'll demonstrate a quick example using our `wmiprvse.exe` scenario. You can find more information about how to use `Adplus` in the Microsoft article at [support.microsoft.com/kb/286350/en-us](http://support.microsoft.com/kb/286350/en-us).

`Adplus` is a `.vbs` script file that comes installed with the Debugging Tools for Windows and must be run from the directory that the debugging tools are installed in (`C:\program files\Debugging Tools For Windows`). When you execute `Adplus` using the correct command-line arguments, the tool will monitor a process and create a dump file when that process spikes the CPU. The following command causes `Adplus` to monitor the `wmiprvse.exe` process and create dump files in the `c:\dumpfiles` directory:

```
C:\Program Files\Debugging
Tools for Windows (x86)>
adplus -hang -pn wmiprvse
.exe -o c:\dumpfiles
```

(The command wraps over several lines because of this magazine's format requirements, but you should type it on one command line.) The `-hang` switch puts `Adplus` into Hang mode, which produces full memory dumps for the process specified on the command line after the script has finished. The `-pn` switch tells `Adplus` which process to monitor. The `-o` switch followed by a folder location tells `Adplus` which directory to store the dump files in.

After the command has run, `Adplus` will automatically create dump files in the specified directory when the process spikes the CPU. If that directory is shared, you can choose to access it from a remote workstation to see whether `Adplus` created any files and review them at your convenience.

You can review the dump files created by `Adplus` for information about runaway processes by following these steps:

1. Run `windbg.exe` from the Debugging Tools for Windows directory. You'll see a screen like that in Figure 2.
2. Click File-Open Crash Dump and point to the directory containing the `Adplus` dump files.
3. Select one of the dump files created by `Adplus`; the file will have a `.dmp` file extension.
4. From the `Windbg` prompt, enter `!runaway`. This command outputs the thread consuming the most CPU time. In our example, this is thread 5.
5. To change the context of the debugger to focus on thread 5, enter the command `~5s`.
6. Now run the `kv` command to get the list of modules and function calls involved in causing the process to spike.

# Runaway Processes

Michael Morales

(Reprinted From WindowsItPro Magazine)

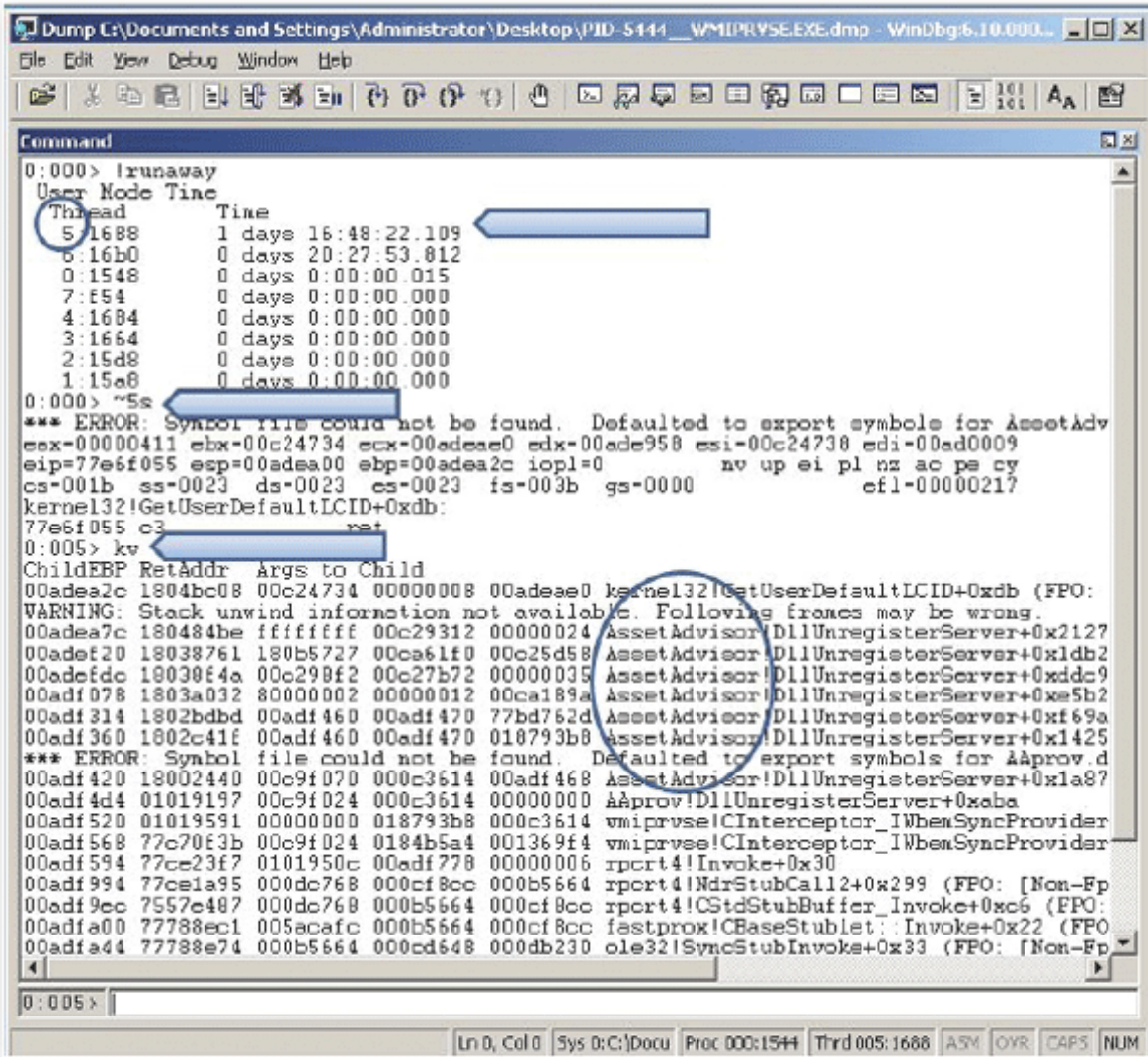


Figure 2: Opening an Adplus dump and running debugging commands

The results of the WinDbg commands you entered in the preceding steps point to the same module that we found to be involved when we used Process Explorer (Asset Advisor.dll).

## More Options

Which tool you use for troubleshooting runaway processes will depend on your circumstances. Process Explorer is easier to use but won't work under certain conditions. Using Adplus involves more steps but is an option when you can't use Process Explorer—and it provides the same troubleshooting information. The next time you have a runaway process bogging down a system, try one or both of these tools, and let me know how they worked for you!