Using the ProcDump Utility to Capture a Wonderware Process Dump

All Tech Notes, Tech Alerts and KBCD documents and software are provided "as is" without warranty of any kind. See the **Terms of Use** for more information.

Topic#: 002732
Created: March 2013

Introduction

ProcDump is a Microsoft command-line utility whose primary purpose is for monitoring an application for CPU spikes and for generating crash dumps during a spike. An administrator or developer can use the generated .dmp file to determine the cause of the spike.

ProcDump also includes hung window monitoring (using the same definition of a window hang that Windows and Task Manager Use), unhandled exception monitoring and it can generate dumps based on the values of system performance counters. It also can serve as a general process dump utility that you can embed in other scripts.

For example, for Wonderware Applications you can use the utility to collect the complete memory when the blue screen appears. This determines which product does this.

You can download the utility from **http://technet.microsoft.com/en-us/sysinternals/dd996900.aspx** (221KB).
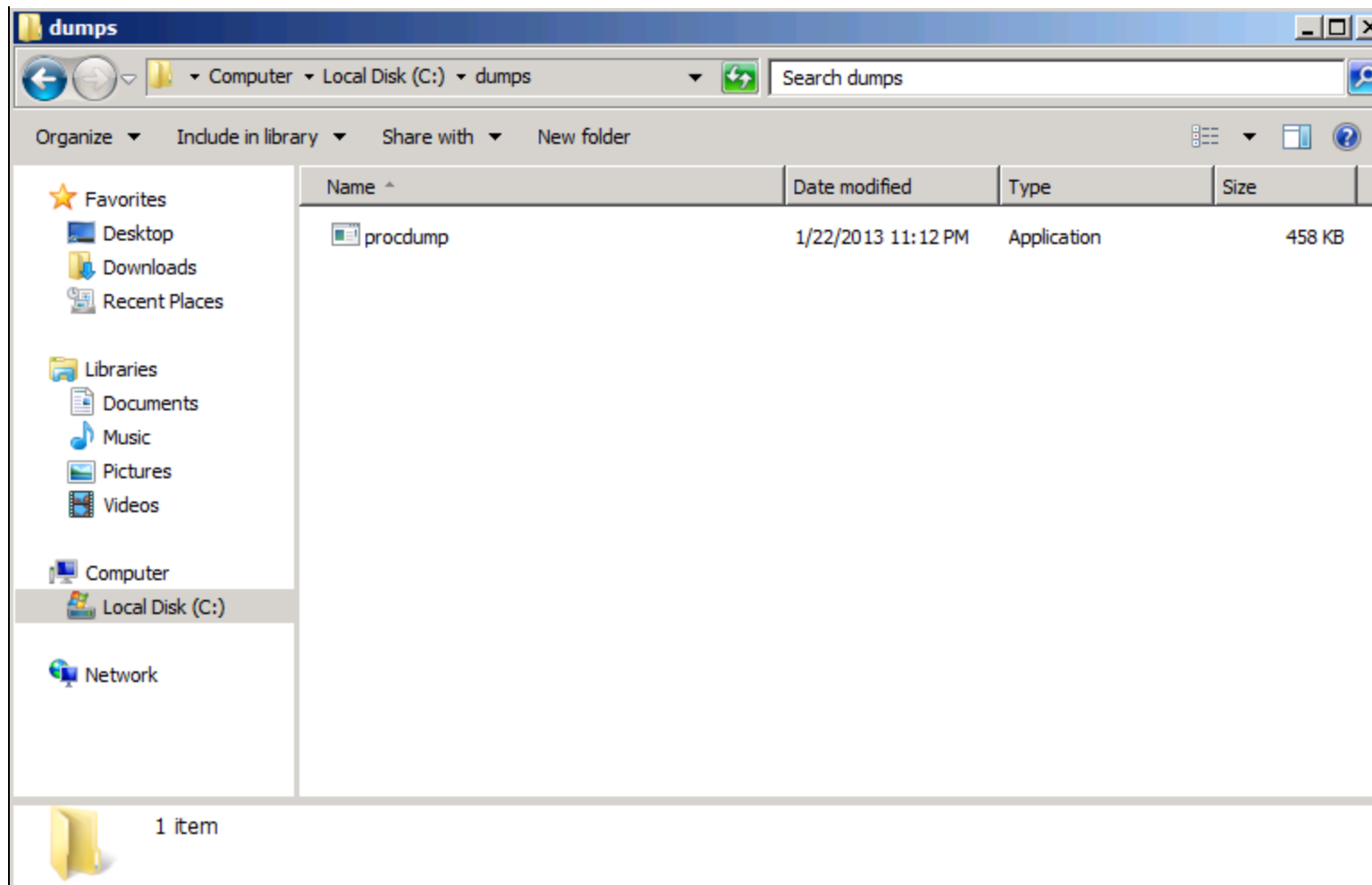
Application Versions

- ProcDump 5.13

- **Client:** Windows XP and higher.

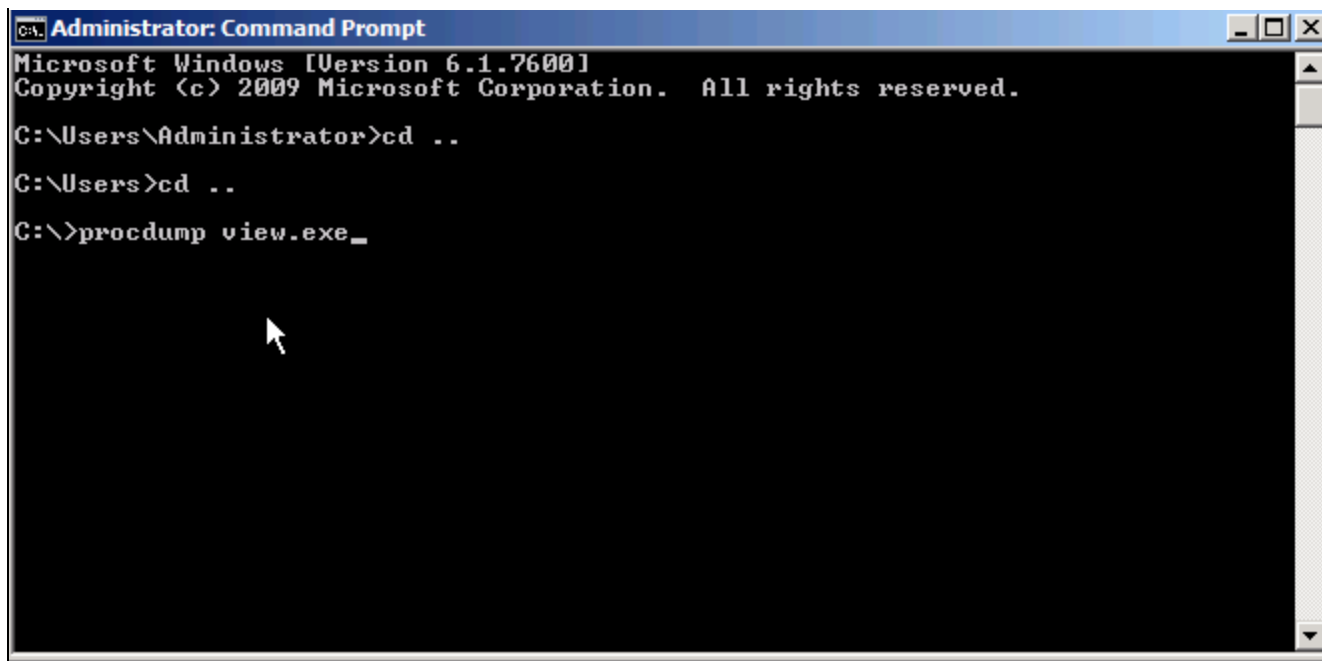- **Server:** Windows Server 2003 and higher.

For the Customer Machine

1. Download and extract the **procdump.exe** file to a designated file on the machine.

   This *Tech Note* uses **C:\dumps**.

2. Open the command prompt.

3. Move the cursor to your C:\ drive (cd) and execute procdump.exe with the required parameters.

FIGURE 2: RUN PROCDUMP UTILITY
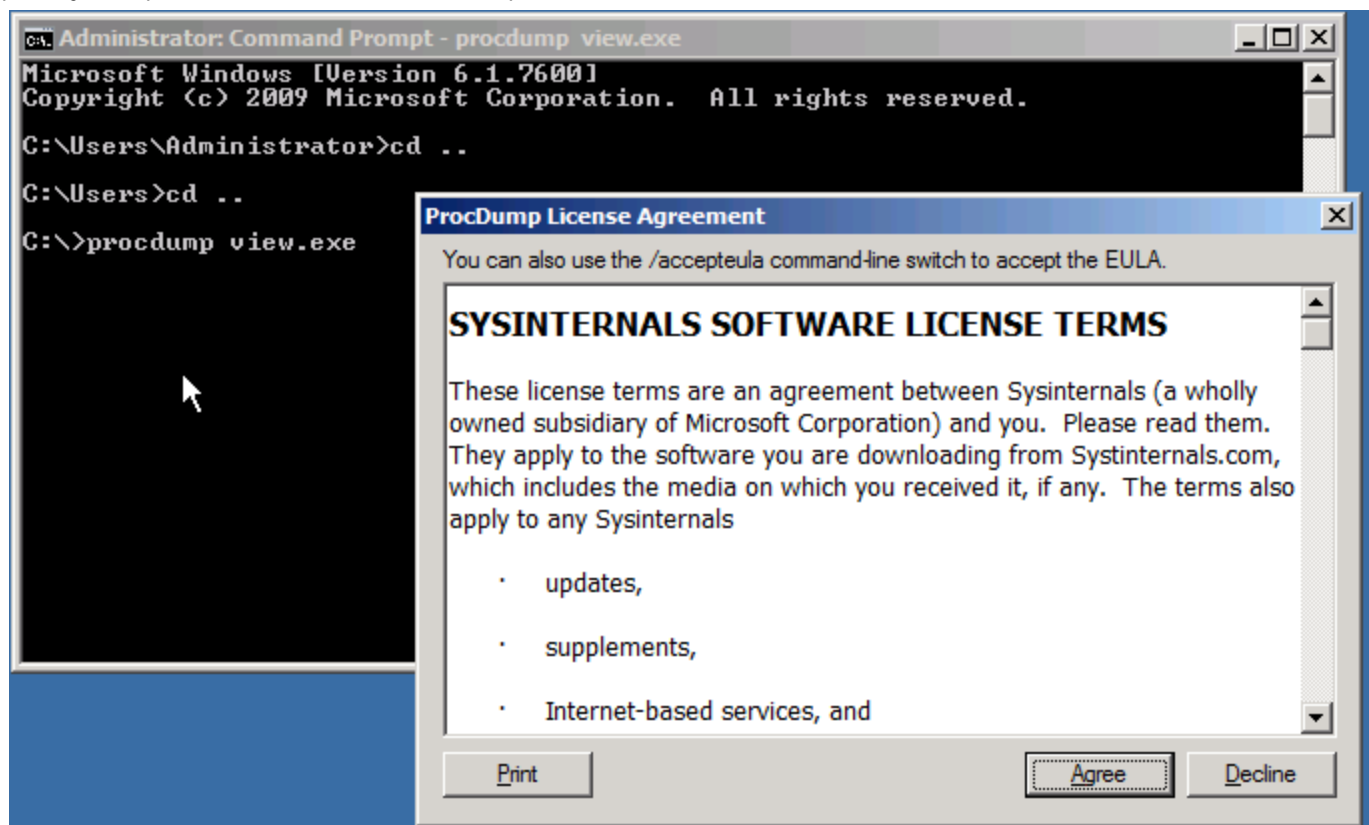
4. Click **Agree**.

FIGURE 3: AGREE TO THE LICENSE AGREEMENT

5. The tool creates the dump file.

6. You can find the dump file on the C:\dumps folder.

Using ProcDump

For complete information on the utility options, click **HERE**.

Wonderware-Specific information is included below.

Examples for Wonderware Applications

- Write 3 mini dumps of a process called **aaEngine.exe** when it exceeds 90% CPU usage for five seconds:

  ```
  C:\>procdump –c 90 –s 5 –n 3 aaEngine.exe
  ```

- Write a mini dump for a process called **view.exe** (InTouch) when one of its Windows is unresponsive for more than 5 seconds:

  ```
  C:\>procdump -h view.exe hungwindow.dmp
  ```

- Write a mini dump for a process called **Historian Index Service** when total system CPU usage exceeds 90% for 10 seconds:

  `C:\>procdump aahIndexSvc -p "\Processor(_Total)\% Processor Time" 90`

- Write a mini dump for **w3wp.exe** (based on IIS Pool, such as WIS Application) if an exception's code/name contains **NotFound**:

  `C:\>procdump -e 1 -f NotFound w3wp.exe`

- Write a full dump of a process with **PID '4572'** (i.e. related to InTouch), and will collect all code/names exceptions (to understand when second exception is required):

  `C:\>procdump -e 1 -f "" 4572`

- Write a full memory dump if the **aaBotstrap.exe** will be terminated within a 1st chance exception (usefully for missing reference exception on services)

  `C:\>procdump -ma aaBotstrap.exe -e 1`

- Write a full memory dump for an **aaEngine.exe** (referenced by his PID 1234) when a crash will be happening, collecting the 2nd chance exception:

  `C:\>procdump -ma 1234 -e 2`

- Write a full memory dump for an **aaEngine.exe** (referenced by hi PID 1234) when a specific exception will be encountered, such as "AccessViolation" 0x000000c most common.

  `C:\>procdump -ma 1234 -e 1 -f "AccessViolation"`

Sending the Dump File for Analysis

Zip/Compress the **.dmp** file and provide it to your Support Engineer.

M. Salah, P. Iodice

For technical support questions, send an e-mail to **wwsupport@invensys.com**.

Back to top