Obtaining a Windows Memory Dump with ProcDump

A White Paper From

For more information, see our web site at
http://www.goldstarsoftware.com

Obtaining a Windows Memory Dump with ProcDump

Last Updated: 08/25/2016

Sometimes, in the course of working a problem with a complex application or service like PSQL, it becomes imperative to capture the state of the process and send it to the software developer for analysis. This paper describes the process of capturing a memory dump of the PSQL database engine with a tool called ProcDump.

Obtain the ProcDump Executable

Download the ProcDump program from Microsoft's web site. At the time of this writing, the current version (v8.0) can be found here:

http://technet.microsoft.com/en-us/sysinternals/dd996900

The name of the file downloaded is currently ProcDump.zip.

Install ProcDump

Open the compressed download file and extract the files PROCDUMP.EXE and PROCDUMP64.EXE to a suitable location on your server. If you want it to be always available, consider moving it to your Windows folder, or some other suitable location in the search path. If you have a 64-bit environment, you want to use ProcDump64.

Start ProcDump to Capture Crashes

Start an Administrative Command Prompt and run ProcDump by itself to get a complete list of options and parameters.

```
C:\ProcDump>procdump

ProcDump v8.0 - Writes process dump files
Copyright (C) 2009-2016 Mark Russinovich
Sysinternals - www.sysinternals.com
With contributions from Andrew Richards

Monitors a process and writes a dump file when the process exceeds the
specified criteria or has an exception.

Capture Usage:
   procdump.exe [-ma | -mp | -d Callback_DLL] [-64]
                [-n Count]
                [-s Seconds]
                [-c|-cl CPU_Usage [-u]]
                [-m|-ml Commit_Usage]
                [-p|-pl Counter_Threshold]
                [-h]
                [-e [1 [-g] [-b]]]
                [-l]
                [-t]
                [-f Filter, ...]
                [-o]
                [-r [1..5] [-a]]
                {
                 {{[-w] Process_Name | Service_Name | PID} [Dump_File | Dump_Folder] }
                |
                 {-x Dump_Folder Image_File [Argument, ...]}
                }
Install Usage:
```

```
   procdump.exe -i [Dump_Folder]
                [-ma | -mp | -d Callback_DLL]
Uninstall Usage:
   procdump.exe -u

   -a      Avoid outage. Requires -r. If the trigger will cause the target
           to suspend for a prolonged time due to an exceeded concurrent
           dump limit, the trigger will be skipped.
   -b      Treat debug breakpoints as exceptions (otherwise ignore them).
   -c      CPU threshold above which to create a dump of the process.
   -cl     CPU threshold below which to create a dump of the process.
   -d      Invoke the minidump callback routine named MiniDumpCallbackRoutine
           of the specified DLL.
   -e      Write a dump when the process encounters an unhandled exception.
           Include the 1 to create dump on first chance exceptions.
   -f      Filter on the content of exceptions and debug logging.
           Wildcards (*) are supported.
   -g      Run as a native debugger in a managed process (no interop).
   -h      Write dump if process has a hung window (does not respond to
           window messages for at least 5 seconds).
   -i      Install ProcDump as the AeDebug postmortem debugger.
           Only -ma, -mp, -d and -r are supported as additional options.
           Uninstall (-u only) restores the previous configuration.
   -l      Display the debug logging of the process.
   -m      Memory commit threshold in MB at which to create a dump.
   -ml     Trigger when memory commit drops below specified MB value.
   -ma     Write a dump file with all process memory. The default
           dump format only includes thread and handle information.
   -mp     Write a dump file with thread and handle information, and all
           read/write process memory. To minimize dump size, memory areas
           larger than 512MB are searched for, and if found, the largest
           area is excluded. A memory area is the collection of same
           sized memory allocation areas. The removal of this (cache)
           memory reduces Exchange and SQL Server dumps by over 90%.
   -n      Number of dumps to write before exiting.
   -o      Overwrite an existing dump file.
   -p      Trigger on the specified performance counter when the threshold
           is exceeded. Note: to specify a process counter when there are
           multiple instances of the process running, use the process ID
           with the following syntax: "\Process(<name>_<pid>)\counter"
   -pl     Trigger when performance counter falls below the specified value.
   -r      Dump using a clone. Concurrent limit is optional (default 1, max 5).
           CAUTION: a high concurrency value may impact system performance.
           - Windows 7   : Uses Reflection. OS doesn't support -e.
           - Windows 8.0 : Uses Reflection. OS doesn't support -e.
           - Windows 8.1+: Uses PSS. All trigger types are supported.
   -s      Consecutive seconds before dump is written (default is 10).
   -t      Write a dump when the process terminates.
   -u      Treat CPU usage relative to a single core (used with -c).
           As the only option, Uninstalls ProcDump as the postmortem debugger.
   -w      Wait for the specified process to launch if it's not running.
   -x      Launch the specified image with optional arguments.
           If it is a Store Application or Package, ProcDump will start
           on the next activation (only).
   -64     By default ProcDump will capture a 32-bit dump of a 32-bit process
           when running on 64-bit Windows. This option overrides to create a
           64-bit dump. Only use for WOW64 subsystem debugging.

Use the -accepteula command line option to automatically accept the
Sysinternals license agreement.

Use -? -e to see example command lines.

If you omit the dump file name, it defaults to <processname>_<datetime>.dmp.
Setting an event with the name "Procdump-<PID>" is the same as typing Ctrl+C
to gracefully terminate Procdump.
```

To launch ProcDump to capture a dump of the database engine, you need to know the name of your engine executable file. For PSQL, this will be one of the following:

Engine Type	**Executable File**
32-bit Server Engine	NTDBSMGR.EXE
64-bit Server Engine	NTDBSMGR64.EXE
Workgroup Engine	W3DBSMGR.EXE

You should then launch ProcDump with the engine executable name, like this:

```
procdump -64 -e -t -ma ntdbsmgr64.exe c:\ntdbsmgr64.dmp
```

Note that the -64 switch is for 64-bit systems – omit this if you are on a 32-bit system. The –e switch is used to create the dump when an unhandled exception is encountered, such as an Access Violation (0xC0000005). The –ma switch dumps all memory, and is usually required to see what is happening inside the engine. Finally, the –t switch causes a dump when the process terminates. The other switches may be useful for capturing other conditions, such as a high CPU usage event.

When ProcDump launches, it will display a screen like this:

```
C:\>procdump -64 -e -ma -t ntdbsmgr64.exe C:\ntdbsmgr64.dmp

ProcDump v8.0 - Writes process dump files
Copyright (C) 2009-2016 Mark Russinovich
Sysinternals - www.sysinternals.com
With contributions from Andrew Richards

Process:               ntdbsmgr64.exe (5076)
CPU threshold:         n/a
Performance counter:   n/a
Commit threshold:      n/a
Threshold seconds:     n/a
Hung window check:     Disabled
Log debug strings:     Disabled
Exception monitor:     Unhandled
Exception filter:      *
Terminate monitor:     Enabled
Cloning type:          Disabled
Concurrent limit:      n/a
Avoid outage:          n/a
Number of dumps:       1
Dump folder:           C:\
Dump filename/mask:    ntdbsmgr64

Press Ctrl-C to end monitoring without terminating the process.
```

Leave this window open in this state until the next crash occurs, and the userdump should be generated in the indicated folder and file name. (If you need to terminate the crash monitoring, simply press Ctrl-C as indicated, and ProcDump will exit.)

Note that the size of the DMP file will depend on how much memory the database engine process is using, so if your L1 cache is set to 8GB, the DMP file will be at least that big.

Before starting this process, be sure that you have sufficient disk space on the drive to which the DMP file will be written, or specify an alternate location on the command line.

Run ProcDump to Capture Engine Memory State Immediately

In certain cases, you may not want to capture a dump when a crash occurs, but rather you want to capture a dump immediately. In this case, you won't be using the flags that wait for an exception, but rather switches to capture an immediate dump, like this:

```
procdump -64 -ma ntdbsmgr64.exe c:\ntdbsmgr64.dmp
```

Submit the Dump to Actian Corporation

If you do not have an active incident opened with Actian, then you will need to either open one or contact Goldstar Software's issue clearinghouse (if previously notified) for the next steps. If you **do** have an active incident opened with Actian, then you can submit your dump directly to them. Zip up the file to shrink it first to reduce transfer time. Be sure to name the ZIP file with the incident number, then upload it to the FTP site below.

Site: ftp2.pervasive.com
User: Pervasive
Password: (Get current password from Actian Support)

Actian changes their FTP software from time to time, you should contact them directly to obtain current FTP login information.

If you still can't get it to work, contact Goldstar Software and let us work with you to help!