**ADPERF: TOOLS: Domain Controller AD Perf Data Collection Script**

Setup For Data Collection:

1. Unzip the PerfScript.zip and place it into a location such as C:\temp. This will include two files. ADDS.xml and a text file ADPerfDataCollection.txt. Please rename the txt file to .ps1.

**\*\*Note:** If there is no ADDS.xml file it will instead use the default AD Data Collector set – You can also provide your own custom xml data collector as long as you name it ADDS.xml and include it in the same directory as the script.

2. Download Procdump found here: <https://docs.microsoft.com/en-us/sysinternals/downloads/procdump>and unzip it to the same directory as above such as C:\temp.

**Note:** If you do not wish to take a procdump of lsass skip this step.

3. **Only For 2012r2 and below:** Install WPR using the following directions:

We will need to download the Windows SDK on

Windows 2008 – 2012r2: <http://msdn.microsoft.com/en-US/windows/desktop/bg162891>

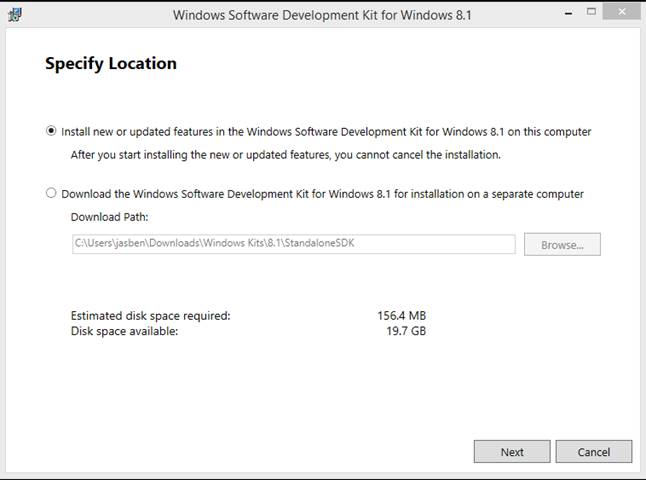
Please scroll down to the section for Earlier Releases and download the appropriate version for your OS:

Windows 8.1 for 2012r2

Windows 8 for 2012

Windows 7 and .Net Framework for 2008r2

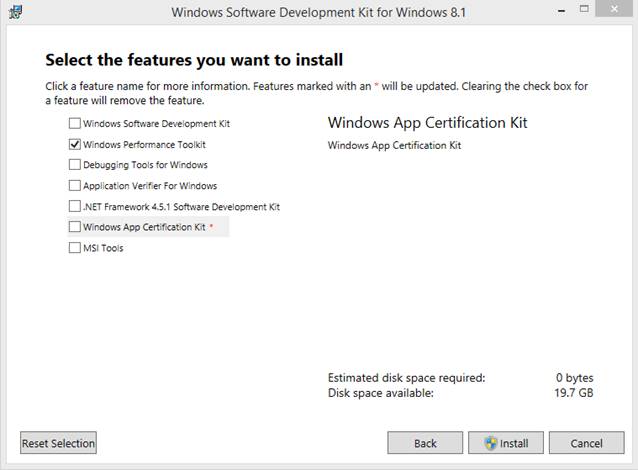
Run the installer and choose “Install new or updated features”



Click Next

Click Accept on the License page

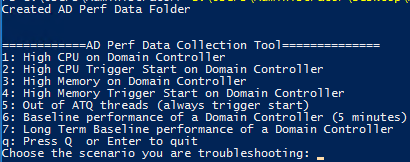
On the “Select the features you want to install” Page, only select “Windows Performance Toolkit”



Click install

**Running The AD Perf Script Collection:**

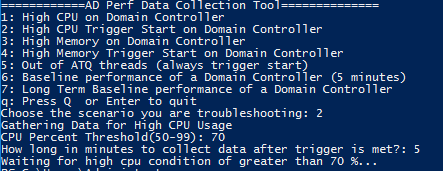
1. Launch the AdPerfDataCollection.ps1 from an **admin** powershell:



2. Choose the appropriate Data Collection scenario for this case- your engineer should also advise which option should be run- If you are not sure, please ask!

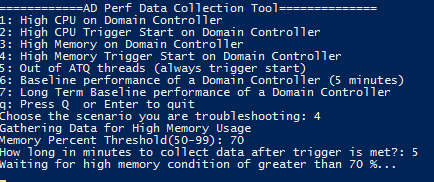
a. Option 1 will enable logging immediately and start data collection. This option should be used if the server is currently in state and will continue to run until you press 'Enter.'

b. Option 2 will trigger based off a CPU % threshold that you set and run for as long as you define between 1-30 minutes (Default is 5). This option should be used for intermittent issue.



c. Option 3 will enable immediately and should be used when the machine is already experience high LSASS Memory usage and will continue to run until you press 'Enter.'

d. Option 4 will trigger based off a Memory % threshold that you set and run for as long as you define between 1-30 minutes (Default is 5). This should be used for intermittent issues or to capture data once in state.



e. Option 5 is a triggered event and will trigger when all ATQ threads are in use.

f. Option 6 is for a baseline run of the data and will begin immediately and stop after 5 minutes by default. Please collect multiple baselines throughout the day for a good sample.

g. Option 7 will collect a long-term baseline. It will collect initial information but then will continue to run just perf counters behind the scene until stopped by the 'Enter' key.

3. After the data collection has finished you may zip up the folder C:\ADPerfData and upload it to the workspace for your case. Please be mindful that the AD Collector Set can take some time to finish on heavily taxed machines. Please wait until the script says it is done before uploading all data… the script will check every 30 seconds for the report.html to create and will let you know when it has finished (It checks for this up to an hour then completes even if the report is not present):

AD%20Perf%20Collection%20Script%20-%20Customer%20Ready_image6.png

Scenario Details:

Outlined below is what each scenario will capture.

**All Scenarios:**

- Enables 1644 Active Directory events with default thresholds

- Enables Netlogon Debug logging with flags 0x2080ffff

- Starts either the custom AD Data Collector Set or the built in Data Collector Set depending on the configuration.

- LSA and LSP tracing

- SamSrv Tracing

- Tasklist /svc and tasklist /v

- Dcdiag report

- Netstat report

- Copies Ntdsai.dll/samsrv.dll/lsasrv.dll/ntdsatq.dll to the central data path

**Scenario 1:**

- Two LSASS process miniplus dumps 5 seconds apart (if procdump is in same directory as script). Runs the following command: procdump.exe lsass.exe -mp -n 2 -s 5 -AcceptEula

- Windows Performance Recorder profiles for GeneralProfile and CPU. Runs the following command: WPR.exe -Start GeneralProfile -Start CPU

**Scenario 2:**

**-** Identical to Scenario 1 but will only start collecting when the configured threshold is met. Uses the performance counter \Processort Information(\_Total)\% Processor Time and checks every 5 seconds.

**Scenario 3:**

- Windows Performance Recorder Profiles for GeneralProfile, Heap, and VirtualAllocation. Runs the following command: WPR.exe -Start GeneralProfile -Start Heap -Start VirtualAllocation

- Collects a single miniplus dump of LSASS. Runs the following command: lsass.exe -mp -AcceptEula

**Scenario 4:**

**-** Identical to Scenario 3 but will only start collecting when the configured threshold is met. Uses the performance counter Memory\% Committed Bytes In Use and checks every 5 seconds

**Scenario 5:**

- Three LSASS process miniplus dumps 5 seconds apart. Runs the following command: procdump.exe lsass.exe -mp -n 3 -s 5 -AcceptEula

**Scenario 6:**

- Enables 1644 Active Directory events with the following thresholds "Search Time Threshold (msecs)" 1, "Expensive Search Results Threshold" 0, "Inefficient Search Results Threshold" 0

- Windows Performance Recorder Profiles for GeneralProfile, CPU, Heap, Virtual Allocation. Runs the following command: WPR.exe -Start GeneralProfile -Start CPU -Start Heap -Start VirtualAllocation

- Three LSASS process miniplus dumps 5 seconds apart. Runs the following command: procdump.exe lsass.exe -mp -n 3 -s 5 -AcceptEula

**Scenario 7:**

**-** Collects a single miniplus dump LSASS. Runs the following command: lsass.exe -mp -AcceptEula

- Enables 1644 Active Directory events with the following thresholds "Search Time Threshold (msecs)" 50, "Expensive Search Results Threshold" 10000, "Inefficient Search Results Threshold" 1000

**-** Enables long and short running perfmon

Logman.exe create counter **PerfLogLong** -o "c:\perflogs\PerfLogLong.blg" -f bincirc -v mmddhhmm -max 300 -c "\LogicalDisk(\*)\\*" "\Memory\\*" "\Cache\\*" "\Network Interface(\*)\\*" "\NTDS(\*)\\*" "\Netlogon(\*)\\*" "\Paging File(\*)\\*" "\PhysicalDisk(\*)\\*" "\Processor(\*)\\*" "\Processor Information(\*)\\*" "\Process(\*)\\*" "\Redirector\\*" "\Server\\*" "\System\\*" "\Server Work Queues(\*)\\*" -si 00:05:00

Logman.exe create counter **PerfLogShort** -o "c:\perflogs\PerfLogShort.blg" -f bincirc -v mmddhhmm -max 300 -c "\LogicalDisk(\*)\\*" "\Memory\\*" "\Cache\\*" "\Network Interface(\*)\\*" "\NTDS(\*)\\*" "\Netlogon(\*)\\*" "\Paging File(\*)\\*" "\PhysicalDisk(\*)\\*" "\Processor(\*)\\*" "\Processor Information(\*)\\*" "\Process(\*)\\*" "\Redirector\\*" "\Server\\*" "\System\\*" "\Server Work Queues(\*)\\*" -si 00:00:05