Update:09/15/2014

Change Lists:

|  |  |  |
| --- | --- | --- |
| Version | Changes | Example |
| 1.2.20 | Added support to extract QDSS from Dump table.  Bug FIxes |  |
| 1.2.19 | Added support for 8992.  FIxed a bug in extracting GCC Reset stats for Bear family.  Added audit for APREG extraction on addresses in DDR  Fixed a bug in populating WARM boot entry counters  Added Support to parse SS dumps based on user request.  Added boot info exception for 8909/8916  8909 IMEM offset corrected  TZ boot info str updated for 64 bit support  TZ Warm boot counters description edited  Status bits from AP\_REG to be displayed for only 8994 currently.  8909 Support added  FIxed an issue where dump tables are corrupted then we display the appropriate error message and exit gracefully |  |
| 1.2.18 | Changes made to get Diag buffer from IMEM. Also Some extra options added to chipinfo structure. This could help save more data specific to chip.  Added explicit garbage collection for files.  Auditing Valid Cores to show up in TZ log  On 8x26, 8x16 – TZ runs out of DDR. SO it is ok to get TZ diag form DDR but for targets, where TZ runs out of IMEM, the Diag would be in OCIMEM (8974, 8994). Changes made to accommodate that. And the pointer to diag buffer is at Shared\_IMEM + 0x720 would point to the Diag location (it could be DDR, or OCIMEM)  Bug Fixes(Bayer’s moore algo) |  |
| 1.2.17 | Added Cab file support – This version will be able to extract files out of cab files and process on it automatically (Supported for only \*.tmp and \*.dmp files).  Support for more than 2 input files. [Supported n < 20]  Changes made to accommodate getting diag buffer form IMEM or DDR based on the chipset. | dexter.exe /v /i1 [file1] /i2 [file2] /i3 [file3] /i4 [file4]… /in [filen] /o c:\temp |
| 1.2.16 | Added tokens to support AP\_REG for 64 bit targets.  Bug fixes. | Dexter.exe [/v] /i FILE [/o] <PATH> /Plat <Target> /Arch <32 or 64> /DTA <DDR Address of DUMP TABLE> /DDRName <DDR FILE NAME>  Please see details about these tokens in later section of this document. |
| 1.2.15 | Added support to look for multiple regions populating in secondary data and getting the last one populating in dump buffer.  Bug Fixes  Added 8994 ID support only. |  |
| 1.2.14 | Bug Fixes  Added post processing for calculating reset reason from retail data buffer |  |
| 1.2.13 | A feature to get Reset and PMIC report from IMEM if SDI executed a reset is added to this release  Added support to get Reset reason from Retail dumps- A new flag added to get the retail reset data form any dump. Please check usage.  Added support for collecting crash time binaries(ex. OCIMEM & QDSS)  Added 8916 support  If device did not boot to OS , then there is a support for getting TZ logs(only if plat form is specified- see release notes- Please check usage) | Dexter.exe [/v] /i FILE [/o] <PATH> /Plat <Target> /GetRetailResetData |
| 1.2.12 | Added a feature to split a hex file based on offsets.  Added Error code entry in Error\_Fatal\_Table  Added Live Dump parsing support | Dexter.exe /s DUMP\_FILE /o1 Begin\_offset(ex:AAAAAAAA) /o2 END\_OFFSET(ex:FFFFFFFF) /c Output\_File\_Name\_with\_PATH  "TZBSP\_ERR\_FATAL\_SMEM",  "TZBSP\_ERR\_FATAL\_XPU\_VIOLATION",  "TZBSP\_ERR\_FATAL\_QSEE" |
| 1.2.11 | Dexter depends on CDI to populate regions data but with New Memory Scheme changes for AP\_REG & QDSS\_ETB\_Trace buffers,CDI does not do that. Dexter changed to work around this change. |  |
| 1.2.10 | Look for RPM's IMem regions in all directories to make sure to add them to Load.cmm |  |
| 1.2.9 | Added support for new Memory Mapping scheme for AP\_REG- the address of buffer is changed to 0x10000 |  |
| 1.2.8 | Added support for loading CodeRAM, DataRam, MessageRAM for EMMC dump as RPM0. RPM1 and RPM2 binaries- sync between windows dump emmc dump.  Added support for new Memory Mapping scheme for AP\_REG and QDSS\_ETB\_trace buffers.  Added FIle logging for SS region info |  |
| 1.2.7 | Retrieves TZ\_LOG from 14C WP8 dmp file. |  |
| 1.2.6 | Bug fix: exception in handling an incomplete dump(no secondary data) |  |
| Generating shared\_imem.bin that T32 script from a meta build can consume. |
| 1.2.5 | TZ warmboot and power collapse counter parsing |  |
| Parsing a standalone tz dump and print out a parsing data to Console | Dexter.exe /t c:\temp\tz\_log0.bin |
| 1.2.4 | Console output for eMMC header information | Dexter.exe /v /e /I c:\temp\input.bin /o . |
| 1.2.2 | TZ log parsing output to result.txt |  |
| 1.2.0 | Bug fix: Find the right start address of the secondary data. |  |
| 1.1.5 | Supporting two input files for SD/USB dump | dexter.exe /v /i1 c:\temp\input1.bin /i2 c:\temp\input2.bin /o c:\temp |
| 1.1.4 | Supporting a network path as an input. | dexter.exe /v /i \\harv-dchung\DumpVerification\DUMP.dmp /o [\\harv-dchung\dropbox](file:///\\harv-dchung\dropbox) |
| Splitter for an option changes from ‘-‘ to ‘/’ |
| Removed checking an an existing folder for an output |  |
| 1.1.2 | Bug fix: Output redirection | Dexter.exe –v –i dump.dmp . > result.txt |

**Dexter.exe**

It is a dump parsing tool for WindowsRT/Windows Phone8. Its main purpose is to extract Qualcomm subsystem dumps out of an input dump file(s). Supported dump file types are:

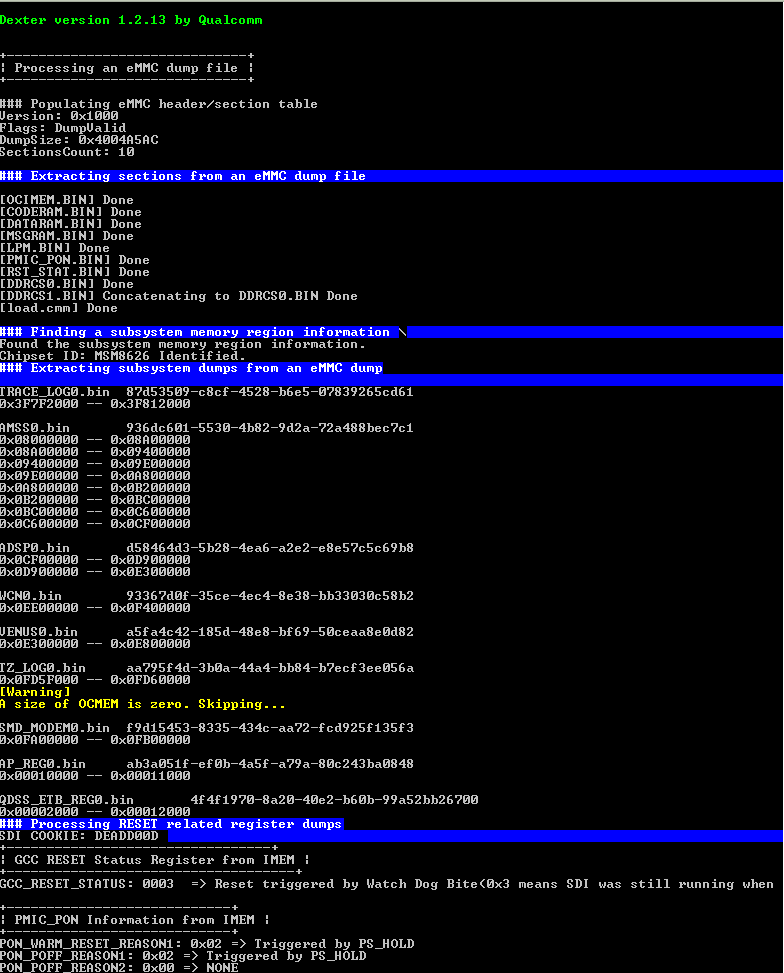
|  |  |  |  |
| --- | --- | --- | --- |
| Dump types | Status | Dependancy | Supported Platform |
| Online Dump(.dmp) | Done | None | Windows RT|Windows Phone 8 |
| Legacy(UEFI) Offline Dump(.dmp) | Done | Raw2dump.exe | Windows Phone 8 |
| eMMC dump(.bin) | Done | None | Windows RT|Windows Phone 8 |
| 14C WP8 dump(.dmp)[[1]](#footnote-1) | Done | Offdumptool.exe | Windows Phone 8 |
| SD/USB dump | Done | None | Windows RT|Windows Phone 8 |
| SSR dump | Done | None | Windows RT|Windows Phone 8 |

This tool is implemented by C# language, thus it does not have any Windbg or Win32 Runtime Library dependancy that the ramdump64.dll has but make sure ***.Net Framework 4.5*** is installed in host PC.

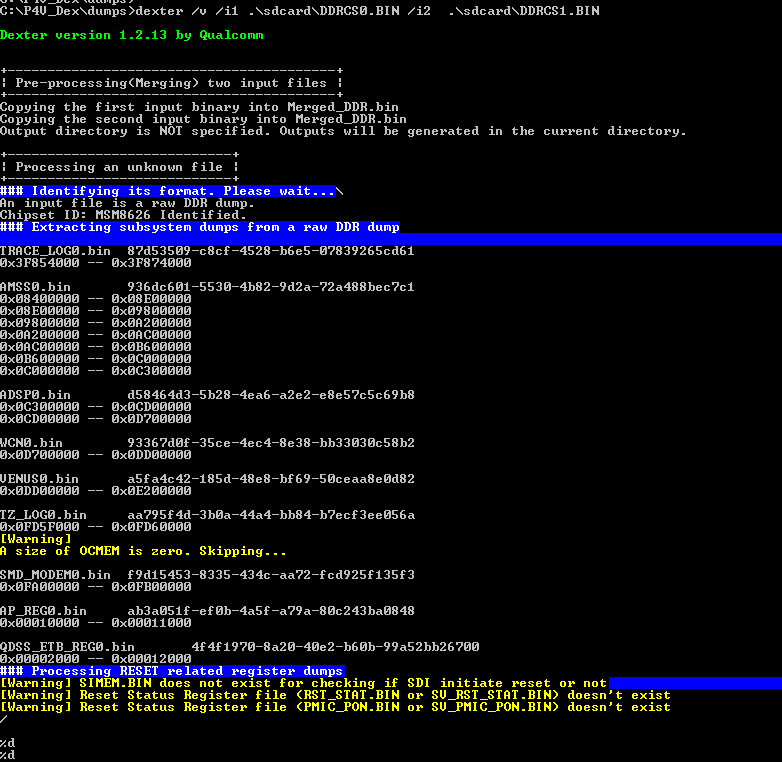
|  |
| --- |
| Usage |
| Dexter.exe /v /i <inputfile> /o <output path> /Plat <Target> /GetRetailResetData  Options:  /v Enabling a verbose mode.  /e Printing eMMC header information to a console output.  /i Specifying an input file.  /o Specifying an output path. If not present, a current working directory will be set to the output path.  /Plat This is optional but user can add target number ex:8974 ,8612, 8626  if plat info is provided, we try to force TZ log buffer to get reset report  /GetRetailResetData This is optional and search the dump for retail reset data if specified  if GetRetailResetData info is provided, we try to force a search for retail reset data and print reset data in results file  /Arch This is used in 64 bit platforms and is necessary to load the correct context.  if Arch is provided we load the AP\_REG context based off its value (32 or 64).  This Is necessary for platforms like 8916 which are supported to run either 32 or 64 bit software.  /DDRName (supported for 8916 and onwards) This is optional and search the file whose name  is given for APREG data only. It assumes the dump file given is DDR dump.  if DDRName info is provided, we use the name of the file given to get APREG info.  /DTA (supported for 8916 and onwards) This is optional and stands for Dump Table  Address. It is address in IMEM.  if DTA info is provide and no IMEM file available but DUMP TABLE DDR address is  known then we can force the program to use the address.  or 64 bit software. |
| Dexter.exe /v /i1 <inputfile1> /i2 <inputfile2> /o <output path> /Plat <Target> /GetRetailResetData  Options:  /v Enabling a verbose mode.  /i1 Specifying the first input file.  /i2 Specifying the second input file.  /in Specifying the nth input file.  /o Specifying an output path. If not present, a current working directory will be set to the output path.  /Plat This is optional but user can add target number ex:8974 ,8612, 8626  if plat info is provided, we try to force TZ log buffer to get reset report  /GetRetailResetData This is optional and search the dump for retail reset data if specified  if GetRetailResetData info is provided, we try to force a search for retail reset data and print reset data in results file  /Arch This is used in 64 bit platforms and is necessary to load the correct context.  if Arch is provided we load the AP\_REG context based off its value (32 or 64).  This Is necessary for platforms like 8916 which are supported to run either 32 or 64 bit software.  /DDRName (supported for 8916 and onwards) This is optional and search the file whose name  is given for APREG data only.It assumes the dump file given is DDR dump.  if DDRName info is provided, we use the name of the file given to get APREG info.  /DTA (supported for 8916 and onwards) This is optional and stands for Dump Table  Address. It is address in IMEM.  if DTA info is provide and no IMEM file available but DUMP TABLE DDR address is  known then we can force the program to use the address. |
| Dexter.exe /t <TZ.bin>  Options:  None. This will automatically set a verbose mode so that console logging is visible by default. |
| Dexter.exe /s DUMP\_FILE /o1 Begin\_offset(ex:AAAAAAAA) /o2 END\_OFFSET(ex:FFFFFFFF) /c Output\_File\_Name\_with\_PATH |

In addition to individual subsystem dumps, the Dexter creates two files; load.cmm and result.txt.

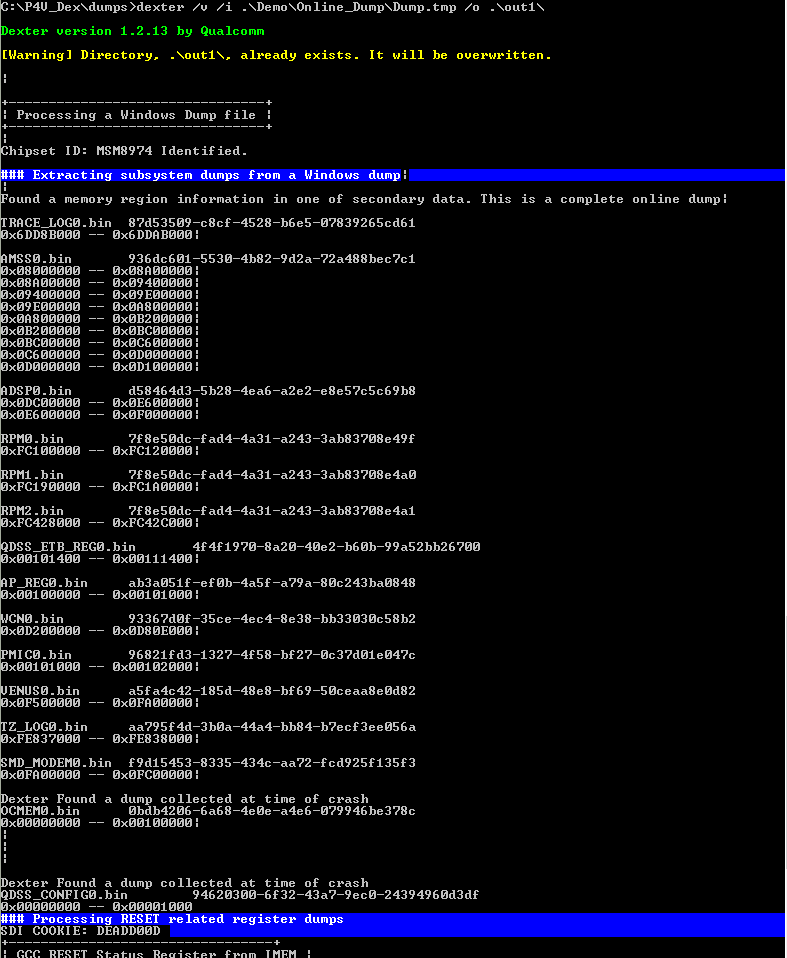
Load.cmm script is used to load captured subsystem binary dumps into a Trace32 and the result.txt contains value of MSM reset status register, PMIC registers, and CPU context information. (Note that the online dump doesn’t provide the CPU context information since OS captures them during bugcheck process)



<figure 1. eMMC dump>



<Figure 2. SD/USB dump>



<Figure 3. Online Dump>

1. Dexter version 1.2.7 and later can retrieve TZ\_LOG information from a 14C WP8 .dmp format. Note that without running Microsoft Offdumptool.exe, Dexter won’t be able to collect other subsystem dumps. [↑](#footnote-ref-1)