KB00115 Terminal Services/Terminal Concentrator Debug Tracing

This article provides an overview of the debug tracing features of Terminal Services and Terminal Concentrator. Normally, trace files are generated for use by QVS for analysis but they can also be useful for integrators and end-users.

The information in this article applies to:

- All versions of Terminal Services
- All versions of Terminal Concentrator

MORE INFORMATION

How to turn on binary tracing to disk

- For TC, tracing can be turned on by creating a LOGGING.ON file in the TC install directory. For Windows TC, this is typically \TC_NT. For 4690 TC, this is \TC46.
- Both TS and TC can be started at a command prompt using 'tprocess -t'. When starting at the command prompt, it is usually best to erase the contents of the TS or TC log directory (e.g. \TSNT\LOG) before starting.
- Another version is 'tprocess -tx' which doesn't start tracing until after the application load is complete.
- For those environments using the CDI configuration file, the keyword traceCmdLine="" can be used to enable binary tracing for both TS and TC, using the -t or the -tx values.
- After the trace data is collected, TC and TS can be shutdown using 'tprocess -k' or 'tprocess -k1'. The -k1 option also forces a .PAN file to be generated and is the recommended way of shutting down a system when collecting data for analysis.
- More information on problem data collection is available in the user manuals.

What is traced?

- Application calls to the operating system. 4690 applications make calls such as open, read, write, get, set, close to what they think is the 4690 operating system. 4690 itself is an extended version of an operating system named FlexOS developed by Digital Research. Historically, many 4690 developers have had access to the FlexOS operating system application developers kit.
- Internal TS/TC subroutine debug tracing.

How tracing works

- Traces are kept on a per-process basis. In a normal or live store environment, tracing is done to wrapping, in-memory trace buffers.
- The traced data is in a binary format to save space and to minimize execution path. Along with the per-process binary information, a system-wide strings table is kept.
- When tracing to disk is enabled, the binary data is flushed to disk when the in-memory trace buffer gets full. The data is saved to TSFn.BIN and TSFn.BI1 files where 'n' is the process sequence number.

How to format trace data

- The binary log files are formatted into ASCII text files by running the e46dumpf program. Generally, the formatter is run against the entire contents of the LOG directory. The ASCII files are named

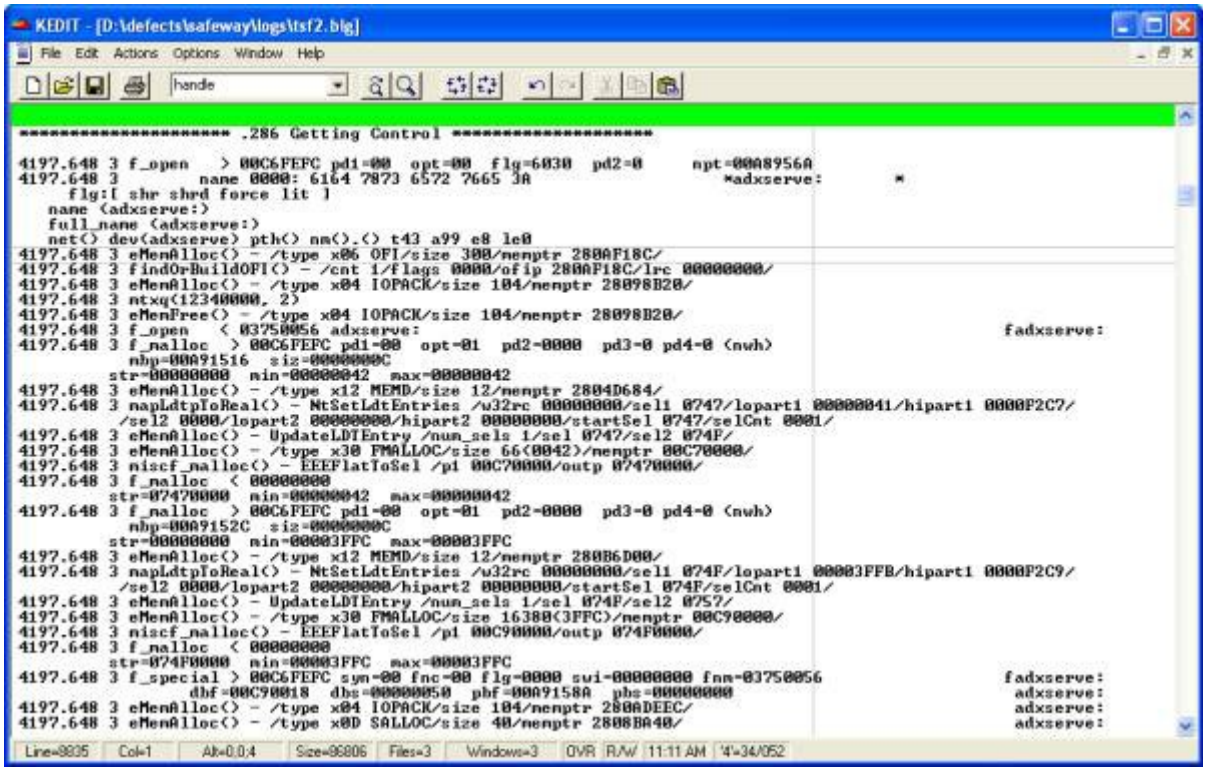
TSFn.BLG, TSFn.BL1, TSFn.BL2, etc. with TSFn.BLG containing the most recent data.

- If 'tprocess -k1' was used to shutdown the system, there will also be a SYSTEM.PAN file (binary) and SYSTEMT.OUT file (formatted ASCII). One of the things in SYSTEMT.OUT will be a list of active processes at the time of shutdown. This list also indicates the sequential process number for each process.

Getting trace information from a .PAN file

- The per-process, in-memory, binary trace buffers and strings table are both included in .PAN files. The formatter finds and formats these buffers into TSFn.PLG files.

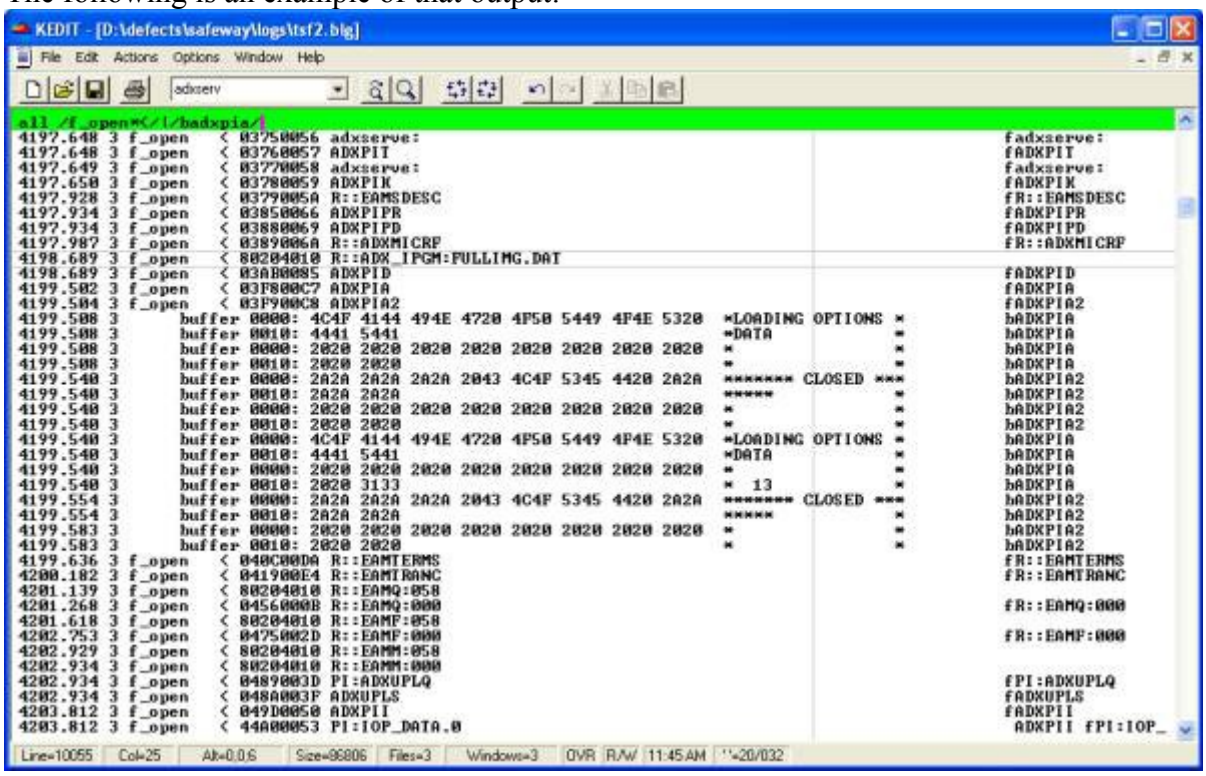
Example formatted trace file:



Tips for understanding trace file information

- FlexOS SVC calls are in the form f_open , f_close, etc. A '>' symbol indicates entry. '<' indicates exit from the function.
- Each timestamp provides millisecond granularity
- The formatter puts filename annotations to the right of many log lines. When the filename is preceded by a 'f', it means the line contains a filenum for that file. When the filename is preceded by a 'b', it means there is a buffer associated with that log line.
- The editor we all use at QVS is something called KEDIT. KEDIT has an extremely powerful 'selective line editing' facility that lets us find useful information in these logs and exclude extraneous information. For example, the KEDIT command all /adxserv causes kedit to remove all lines from view that do not include the text "adxserv." Furthermore, KEDIT allows its all targets to be expressions such as all /f_open*</ | /badxpia/. This example will find lines containing 'f_open' followed by '<' (with anything in-between) OR lines containing 'badxpia' and show only those lines.

The following is an example of that output:




```
4197.648 3 f_open    < 03750056 adxserve:                                                    fadxserve:
4197.648 3 f_open    < 03760057 ADXPIT                                                       fADXPIT
4197.649 3 f_open    < 03770058 adxserve:                                                    fadxserve:
4197.650 3 f_open    < 03780059 ADXPIK                                                       fADXPIK
4197.928 3 f_open    < 0379005A R::EAMSDESC                                                  fR::EAMSDESC
4197.934 3 f_open    < 03850066 ADXPIPR                                                      fADXPIPR
4197.934 3 f_open    < 03880069 ADXPIPD                                                      fADXPIPD
4197.987 3 f_open    < 0389006A R::ADXMICRP                                                  fR::ADXMICRP
4198.689 3 f_open    < 80204010 R::ADX_IPGM:FULLIMG.DAT
4198.689 3 f_open    < 03AB0085 ADXPID                                                       fADXPID
4199.502 3 f_open    < 03F800C7 ADXPIA                                                       fADXPIA
4199.504 3 f_open    < 03F900C8 ADXPIA2                                                      fADXPIA2
4199.508 3       buffer 0000: 4C4F 4144 494E 4720 4F50 5449 4F4E 5320  *LOADING OPTIONS *    bADXPIA
4199.508 3       buffer 0010: 4441 5441                                *DATA             *    bADXPIA
4199.508 3       buffer 0000: 2020 2020 2020 2020 2020 2020 2020 2020  *                 *    bADXPIA
4199.508 3       buffer 0010: 2020 2020                                *                 *    bADXPIA
4199.540 3       buffer 0000: 2A2A 2A2A 2A2A 2043 4C4F 5345 4420 2A2A  ******* CLOSED ***    bADXPIA2
4199.540 3       buffer 0010: 2A2A 2A2A                                *****             *    bADXPIA2
4199.540 3       buffer 0000: 2020 2020 2020 2020 2020 2020 2020 2020  *                 *    bADXPIA2
4199.540 3       buffer 0010: 2020 2020                                *                 *    bADXPIA2
4199.540 3       buffer 0000: 4C4F 4144 494E 4720 4F50 5449 4F4E 5320  *LOADING OPTIONS *    bADXPIA
4199.540 3       buffer 0010: 4441 5441                                *DATA             *    bADXPIA
4199.540 3       buffer 0000: 2020 2020 2020 2020 2020 2020 2020 2020  *                 *    bADXPIA
4199.540 3       buffer 0010: 2020 3133                                *  13             *    bADXPIA
4199.554 3       buffer 0000: 2A2A 2A2A 2A2A 2043 4C4F 5345 4420 2A2A  ******* CLOSED ***    bADXPIA2
4199.554 3       buffer 0010: 2A2A 2A2A                                *****             *    bADXPIA2
4199.583 3       buffer 0000: 2020 2020 2020 2020 2020 2020 2020 2020  *                 *    bADXPIA2
4199.583 3       buffer 0010: 2020 2020                                *                 *    bADXPIA2
4199.636 3 f_open    < 040C00DA R::EAMTERMS                                                  fR::EAMTERMS
4200.182 3 f_open    < 041900E4 R::EAMTRANC                                                  fR::EAMTRANC
4201.139 3 f_open    < 80204010 R::EAMQ:058
4201.268 3 f_open    < 0456000B R::EAMQ:000                                                  fR::EAMQ:000
4201.618 3 f_open    < 80204010 R::EAMF:058
4202.753 3 f_open    < 0475002D R::EAMF:000                                                  fR::EAMF:000
4202.929 3 f_open    < 80204010 R::EAMM:058
4202.934 3 f_open    < 80204010 R::EAMM:000
4202.934 3 f_open    < 0489003D PI:ADXUPLQ                                                   fPI:ADXUPLQ
4202.934 3 f_open    < 048A003F ADXUPLS                                                      fADXUPLS
4203.812 3 f_open    < 049D0050 ADXPII                                                       fADXPII
4203.812 3 f_open    < 44A00053 PI:IOP_DATA.0                                                 ADXPII fPI:IOP_
```


- Some 4690 device filename information: ANDISPLAY: is ADXPIA, ANDISPLAY2: is ADXPIA2, CDRAWER: is ADXPID, IOPROC: is ADXPII, MSR: is ADXPIM, CR: is ADXPIPR, DI: is ADXPIPD, TOTRET: is ADXPIT.