OpenQM Application Event Profiling

https://www.openqm.com/support/documentation/

The PROFILER tool help developers track down application performance issues by recording key events such as opening a file and logging how often and where these occur. The events recorded are:

Event	Description
CALL	Subroutine calls.
OPEN	OPEN and OPENPATH statements.
READ	READ and MATREAD statements plus locking variations.
WRITE	WRITE and MATWRITE statements.
DELETE	DELETE statements.
OPENSEQ	OPENSEQ statements.
READSEQ	READSEQ statements.
WRITESEQ	WRITESEQ statements.
EXECUTE	EXECUTE statements.
OBJECT	Calls to public subroutines and functions in a class module.

The PROFILER Command

Event logging is started with

```
> PROFILER ON USER n
```

If the user option is omitted, profiling is enabled for the process in which the command is used.

Profiling data is collected in memory until it is disabled with

```
> PROFILER OFF USER n
```

Disabling profile data collection also writes the data to disk as a file with a name of the form profile.123 in the QM temporary directory where 123 is the user number.

The third mode of the PROFILER command is to display the data:

```
> PROFILER DISPLAY USER n
```

If none of ON, OFF and DISPLAY is present, DISPLAY is assumed.

The first screen shown by in display mode is a list of the available profile log files:

```
Profiler log files
 Logs     Date/time
>11       17 Jun 20 12:33:20
 14       18 Jun 20 14:21:07
```

The ">" line selection marker can be moved with the usual cursor positioning keys to select the file to be processed. Pressing the Return key moves to a display of the total number of logged events of each type:

```
Item types present in log for user 11
 Counts   Item Types
>606      Call
 20       Execute
 40       Open
 1        OpenSeq
 2157     Read
 45       Write
```

Again, navigate to the type of interest and press Return to see more detail.

For Call events, the display shows the names and call counts for each logged subroutine call.

```
Subroutine call counts
 Counts   Subroutine Name
>320      !PARSER
 100      $QDISP
 1        ASSIGN.CALL
 46       TIMESTAMP
```

Names starting with a $ symbol are internal components of QM.

Diving into an entry from the list of calls shows a further breakdown of where the calls occurred.

```
Callers of TIMESTAMP
 Counts   Calling Program
>2        ASSIGN.CALL
 17       CALL.STATUS
 1        CLOSE.CALL
 3        UPDATE.CALL
```

A final level of detail, available only for programs compiled in debug mode, shows the line numbers at which the calls occurred.

```
Calls to TIMESTAMP from UPDATE.CALL
 Counts   Line Number
>2        24
 1        17
```

Similar drill-down layers are used for other event types:

Event	Description
CALL	Subroutine name Called from
OPEN	File name Opening program name
READ WRITE DELETE	File name Program name
OPENSEQ	File name Opening program name
READSEQ WRITESEQ	File name Program name
EXECUTE	Executed command (first 32 characters) Origin program name
OBJECT	Object class name Called from

In all cases, the line number information is available for debug mode programs.

Navigation

Return		Dive to next level
Backspace		Up to previous level
Cursor up	P U	Move selection up one line
Cursor down	N D	Move selection down one line
Page up		Move selection to bottom of previous page
Page down		Move selection to top of next page
Home	T	Move selection to top of first page
End	B	Move selection to bottom of last page
Cursor left	L	Pan left
Cursor right	R	Pan right
	S	Toggle sort order
	Q	Quit
	?	Show help screen