

# Table of Contents

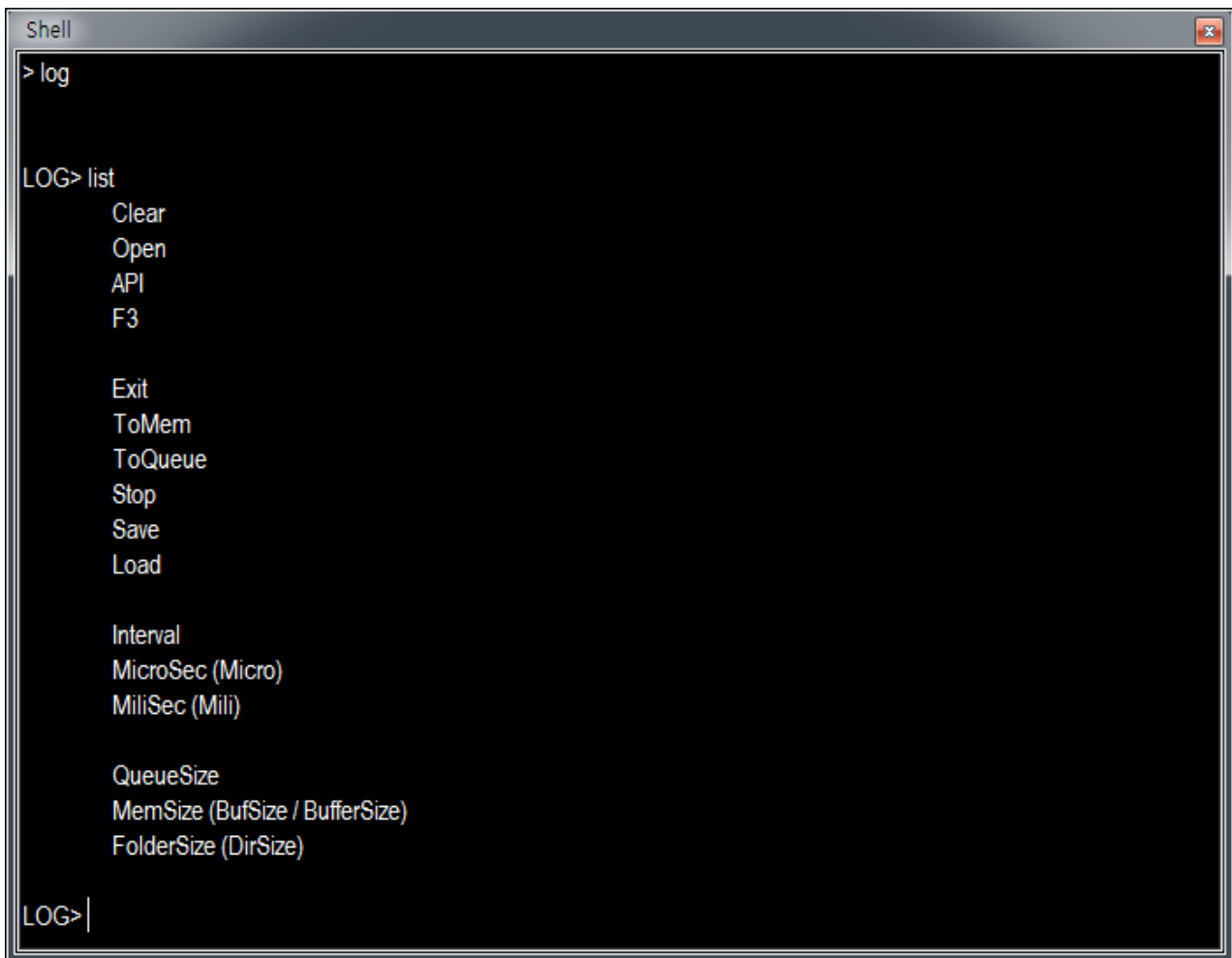
Intro .....	1
Logging .....	1
Interval .....	3
LogType .....	4

# LogMode

× API를 반복실행하거나 로깅할 수 있는 LogMode에 대한 안내 페이지입니다.

## Intro

- LogMoed에서는 API 실행 시 Stop 명령이 입력될때까지 API를 반복 호출하고, 그 결과<sup>1)</sup>를 .txt 파일로 기록합니다.
- 'Log' 입력 시 LogMode로 변경됩니다.
- 명령 입력 시 대문자 / 소문자는 구분하지 않습니다.



```
Shell
> log

LOG> list
  Clear
  Open
  API
  F3

  Exit
  ToMem
  ToQueue
  Stop
  Save
  Load

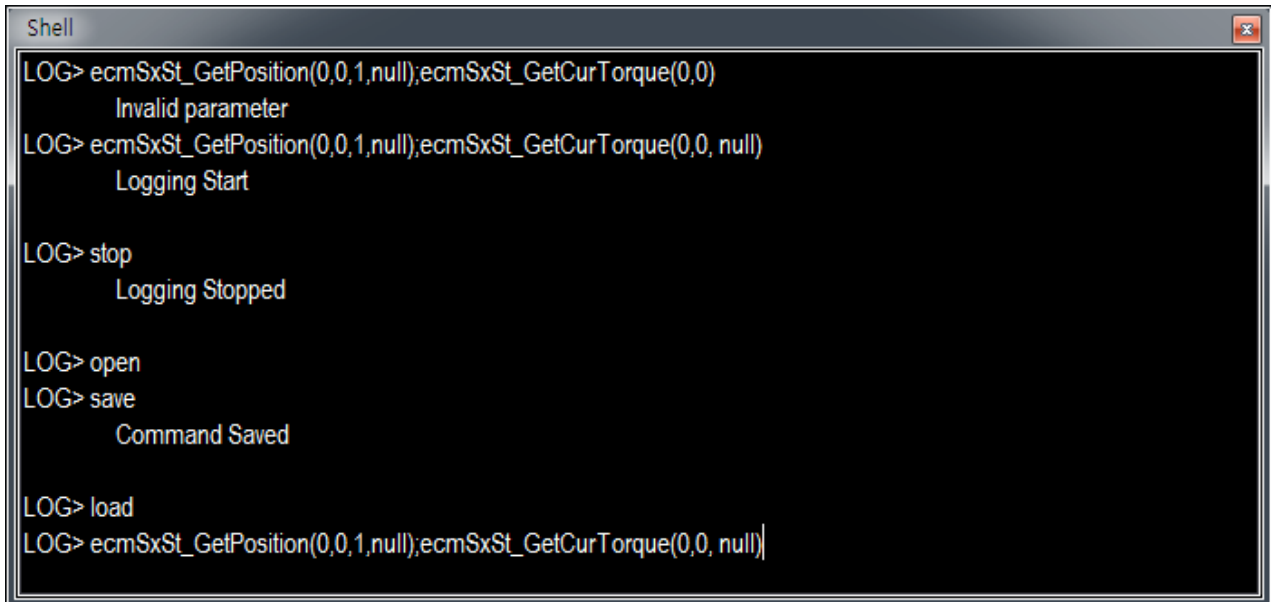
  Interval
  MicroSec (Micro)
  MiliSec (Mili)

  QueueSize
  MemSize (BufSize / BufferSize)
  FolderSize (DirSize)

LOG> |
```

## Logging

- API 단위 입력과 입력 방식이 같습니다.
- ';'으로 여러개의 API를 연결할 수 있습니다.
- out type parameter에 “null” 을 입력하면, 해당 파라메타 결과값은 logging 되지 않습니다.
- 잘못된 parameter 입력 시 “Invalid parameter”를 리턴하며, 정상적으로 logging이 시작되면 “Logging Start”를 출력합니다.



```

Shell
LOG> ecmSxSt_GetPosition(0,0,1,null);ecmSxSt_GetCurTorque(0,0)
    Invalid parameter
LOG> ecmSxSt_GetPosition(0,0,1,null);ecmSxSt_GetCurTorque(0,0, null)
    Logging Start

LOG> stop
    Logging Stopped

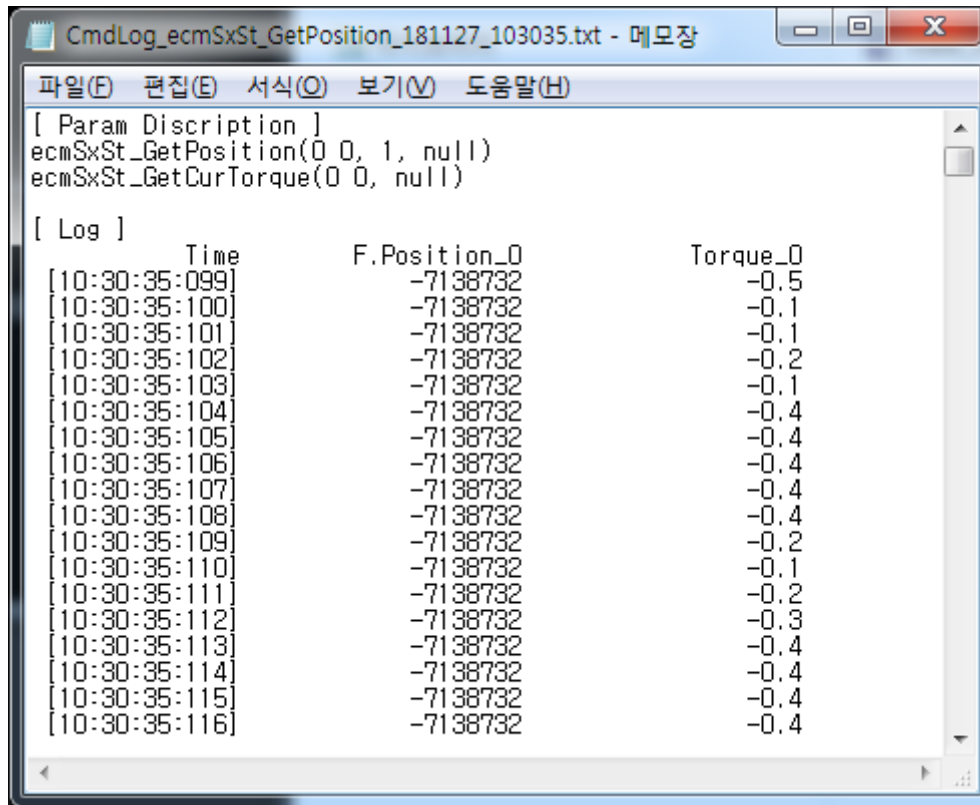
LOG> open
LOG> save
    Command Saved

LOG> load
LOG> ecmSxSt_GetPosition(0,0,1,null);ecmSxSt_GetCurTorque(0,0, null)
  
```

```

LOG> stop
LOG> open
LOG> save
LOG> load
  
```

- stop : logging을 종료합니다. 종료 시 “Logging Stopped”를 출력합니다.
- open : logFile이 저장된 디렉토리를 불러옵니다.
- save : 마지막 로깅 명령을 저장합니다.
- load : 저장된 명령을 불러옵니다.
- exit : LogMode를 종료하고 NormalMode 로 돌아갑니다.



### Interval

- Logging 간격을 설정할 수 있습니다.

```
Shell
> log

LOG> interval
    Interval = 10 ms

LOG> interval(1)
    Log Interval (ms) : 10 → 1

LOG> microsec
    Resolution Changed : MiliSecond → MicroSecond

LOG> interval
    Interval = 1000 us

LOG> interval(500)
    Log Interval (us) : 1000 → 500

LOG> milisec
    Resolution Changed : MicroSecond → MiliSecond

LOG> interval
    Interval = 1 ms

LOG>
```

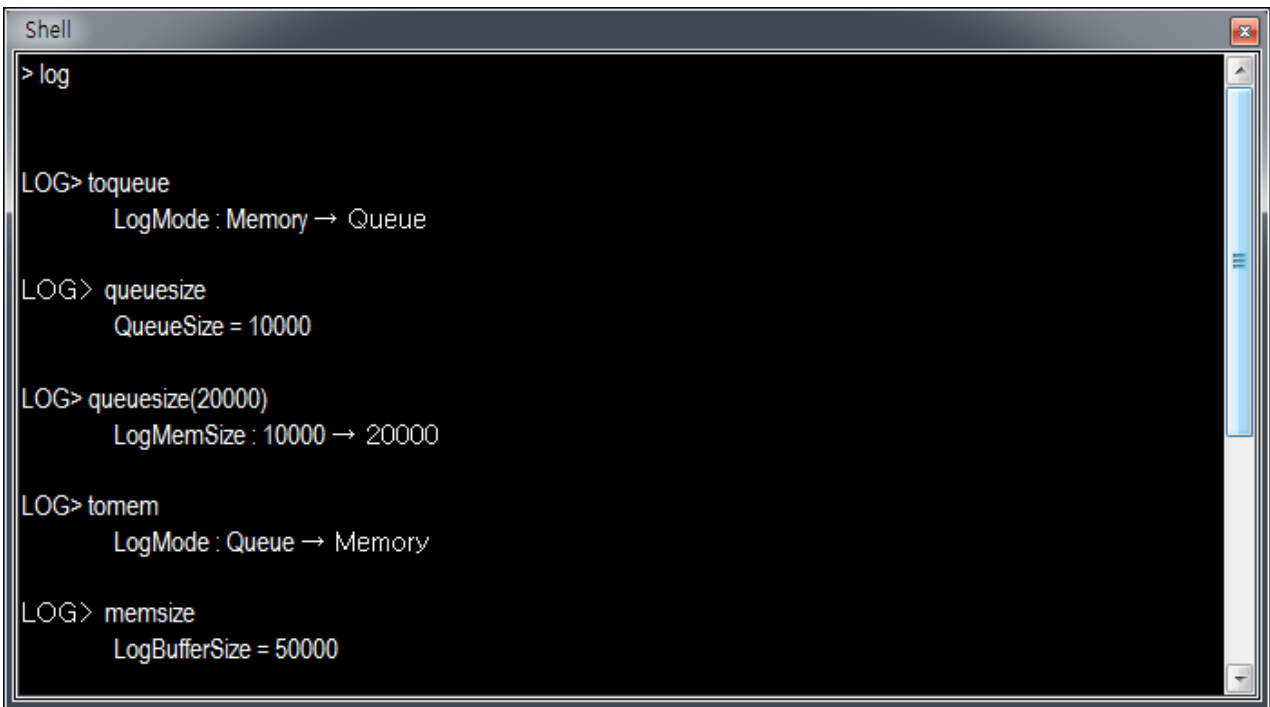
```
LOG> interval
LOG> interval(1)
LOG> microsec
LOG> interval
LOG> interval(500)
LOG> milisec
```

- interval : log 주기를 보여줍니다. 기본값은 10ms 입니다.
- interval(1) : log 주기를 1ms 로 변경합니다.
- microsec : log 주기의 단위를 us 로 변경합니다.
- interval : microsec 주기와 milisec 주기는 별도로 관리됩니다.
- interval(500) : microsec 단위일때는 microsec 단위 주기가 변경됩니다.

× logging 주기는 pc 성능 및 사용량에 따라 일부 불규칙할 수 있습니다.  
특히, microsec 단위 로깅은 pc 성능에 크게 영향을 미치므로 권장하지 않습니다.

## LogType

- Memory 방식과 Queue 방식을 지원합니다.



```

Shell
> log

LOG> toqueue
    LogMode : Memory → Queue

LOG> queuesize
    QueueSize = 10000

LOG> queuesize(20000)
    LogMemSize : 10000 → 20000

LOG> tomem
    LogMode : Queue → Memory

LOG> memsize
    LogBufferSize = 50000
  
```

## Memory

- MemSize 만큼 Memory에 로깅 후 파일로 기록합니다.
- 더블버퍼링으로 운용되며, Stop 명령시까지 계속 파일이 생성됩니다.

## Queue

- QueueSize 만큼만 로깅되며, stop 명령 시 하나의 파일만 생성됩니다.

```

LOG> toqueue
LOG> quesize
LOG> quesize(20000)
LOG> tomem
LOG> memsize
LOG> memsize(20000)
LOG> dirsiz
LOG> dirsiz(1000)
  
```

- toqueue : Queue 방식으로 변경합니다.
- quesize : queueSize를 확인합니다. 단위는 line입니다.
- quesize(20000) : quesize를 20000 으로 변경합니다.
- tomem : Memory 방식으로 변경합니다. Shell 실행 시 기본값입니다.
- memsize : MemorySize를 확인합니다. 단위는 line입니다.

- memsize(20000) : MemorySize를 20000으로 변경합니다.
- dirsize : log file이 생성되는 폴더의 크기제한값을 확인합니다. dirSize 이상으로 폴더 크기가 커지는 경우, 가장 오래된 file부터 삭제됩니다.
- dirsize(2048) : dirSize를 2048로 변경합니다. 단위는 MB입니다.

1)

Return 값과 out 파라메타

From:

<http://comizoa.com/info/> - -

Permanent link:

[http://comizoa.com/info/doku.php?id=application:comiide:tool:shell:shell\\_logmode&rev=1559007493](http://comizoa.com/info/doku.php?id=application:comiide:tool:shell:shell_logmode&rev=1559007493)

Last update: **2024/07/08 18:23**