# Debug commands

## Enable Voximal module traces

Execute :

```
CLI> voximal debug
```

## Enable Voximal interpreter traces

Execute :

```
CLI> voximal debug interpreter
```

## View the traces flow

Execute :

```
root# tail -f /var/log/voximal/debug.log
```

With colors :

```
root# tail -f /var/log/voximal/debug.log | ccze -A
```

## Generate a normal stop

Clean stop of the Voximald process :

```
root# /usr/sbin/voximalc -local 2 -distant 1 -mode 2

Command-line arguments :
[-key id] [-local id] [-distant id] [-send message] [-mode 0..3]
To send one message, set mode to 0
To receive messages in loop, set mode to 1
To work in request/response, set mode to 2
To send messages in loop, set mode to 3

> exit
< exit|result=ok
```

## Powerful top monitor

Use the “htop” tool :

```
htop -p $(pidoff voximald)
```

```
  1  [                          0.0%]   Tasks: 33, 29 thr; 1 running
  2  [|                         0.7%]   Load average: 0.51 0.28 0.30
  3  [                          0.0%]   Uptime: 3 days, 01:39:39
  4  [                          0.0%]
  Mem[|||                 175M/3.91G]
  Swp[|                  31.8M/7.81G]

  PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
    1 root       20   0  199M  3316  2208 S  0.0  0.1  0:02.58 /sbin/init
18890 root       20   0 56440  4164  3360 S  0.0  0.1  0:00.02 `-
/lib/systemd/s
18891 root       20   0  224M  1316     0 S  0.0  0.0  0:00.00 |  `- (sd-
pam)
  409 root       20   0 81156  1236  1068 S  0.0  0.0  0:00.95 `-
/usr/lib/postf
14487 postfix    20   0 83224  5088  4320 S  0.0  0.1  0:00.00 |  `- pickup
-l -
  412 postfix    20   0 83388  1916  1648 S  0.0  0.0  0:00.30 |  `- qmgr -l
-t
  308 root       20   0  350M 24556 19196 S  0.0  0.6  0:05.93 `-
/usr/sbin/apac
30788 www-data   20   0  350M  6752  1384 S  0.0  0.2  0:00.00 |  `-
/usr/sbin/a
30787 www-data   20   0  350M  6752  1384 S  0.0  0.2  0:00.00 |  `-
/usr/sbin/a
30786 www-data   20   0  350M  6752  1384 S  0.0  0.2  0:00.00 |  `-
/usr/sbin/a
30785 www-data   20   0  350M  6752  1384 S  0.0  0.2  0:00.00 |  `-
/usr/sbin/a
30784 www-data   20   0  350M  6752  1384 S  0.0  0.2  0:00.00 |  `-
/usr/sbin/a
  263 mysql      28   8  670M 38608  1668 S  0.0  0.9  1:55.94 `-
/usr/sbin/mysq
 2808 mysql      20   0  670M 38608  1668 S  0.0  0.9  0:00.00 |  `-
/usr/sbin/m
F1Help  F2Setup F3SearchF4FilterF5SortedF6CollapF7Nice -F8Nice +F9Kill
F10Quit
```

## Powerful SIP traffic monitor

Use the “sngrep” tool :

```
sngrep
```

```
                                      xINVITE
sip:0970265644@84.246.228.243:506
```

```
           128.140.150.200:5060       xuser=phone SIP/2.0
          qqqqqqqqqqwqqqqqqqq         xVia: SIP/2.0/UDP
128.140.150.200:5060;br
  22:22:36.632419   x        INVITE (Sxch=z9hG4bKk5efg100aob0k0be5og0.1
        +0.000994   x qqqqqqqqqqqqqqqqxMax-Forwards: 66
  22:22:36.633413   x         100 TryixFrom:
<sip:0677379042@172.17.28.229;user
        +0.048311   x <qqqqqqqqqqqqqqqxhone>;tag=1c1397877903
  22:22:36.681724   x        200 OK (SxTo: sip:0970265644@ipdirections.net
        +0.085629   x <qqqqqqqqqqqqqqqxCall-ID: SDk6upe02-
efa980599ffc4b83e9221
  22:22:36.767353   x             ACK xec76ea708-v300g00030
        +8.759074   x qqqqqqqqqqqqqqqxCSeq: 1 INVITE
  22:22:45.526427   x             BYE xContact:
<sip:128.140.150.200:5060;maddr
        +0.000527   x qqqqqqqqqqqqqqqx28.140.150.200>
  22:22:45.526954   x           200 OKxSupported: sdp-anat
                    x <qqqqqqqqqqqqqqqxAllow: INVITE,ACK,CANCEL,BYE,OPTIONS
                    x                 xP-Asserted-Identity:
<sip:0677379042@172
                    x                 x7.28.229;user=phone>
                    x                 xContent-Type: application/sdp
                    x                 xContent-Length: 435
                    x                 xP-Early-Media: supported
                    x                 xP-Access-Network-Info: GSTN;operator-
spe
                                      x
Esc Calls List   Enter Raw   Space Compare   F1 Help   F2 SDP   F3 RTP   F4
Exte
```

## Asterisk Manager traffic monitor

Use the “ngrep” tool :

```
ngrep
```

```
ngrep -d lo -s 1500 port 5038 -T
```

## Monitor memory/CPU

A very simple way to follow the memory and CPU indicators :

```
root# top -b -d 5 -p $(pidof voximald) | awk -v OFS="," '$1+0>0 {print
strftime("%Y-%m-%d %H:%M:%S"),$1,$NF,$5,$6,$7,$9,$10; fflush() }' | tee
datas.csv
```

## Valgrid using

The Asterisk module launch the interpreter by defaut, you need to disable it to be able to run the voximald process with Valgrid.

To disable the Voximal launch from the Asterisk module edit and add this option in the voximal.conf :

```
[general]
...
launcher=no
...
```

Start the voximald process with Valgrid :

```
#root valgrind  --tool=memcheck --leak-check=full --log-file="logfile.out"
/usr/sbin/voximald -channels 1 -config /etc/openvxi/client.cfg -user
asterisk -group asterisk
```

And run the Asterisk.

```
asterisk -cvvvvv -U asterisk -G asterisk -g
```

## VoiceXML Log levels

Format :

```
  Diagnostic
  date/time | threadID | sessionID or mark | tagID | subtag | text
  Error
  date/time | threadID | sessionID or -1 | 0 | severity | modulename |
errorID | errorText | appends
  Event
  date/time | threadID | sessionID | EVENT | 0|evenID | appends
```

Logging example :

```
...
Oct 23 23:39:12.35|0x7f82daebb700|-1|4002||AccessBegin Locked
Oct 23 23:39:12.35|0x7f82daebb700|-1|4002||AccessEnd Unlock
Oct 23
23:39:12.35|0x7f82daebb700|0_1_1540337949.313|4000|SBjsiGetVar|exiting:
returned 0, 0x7f82daeb34e0 (0x7f82d419a750)
Oct 23
23:39:12.35|0x7f82daebb700|0_1_1540337949.313|8002|fr.ulex.vxi|VXI::assign_e
lement(set value : id=08facd13i9e0q9o1m4vav8ghvn8tbagn21bfcb5)
Oct 23
23:39:12.35|0x7f82daebb700|0_1_1540337949.313|5001|VXIpromptWait|VXIprompt
```

```
Oct 23
23:39:12.35|0x7f82daebb700|0_1_1540337949.313|5001|VXIpromptWait|Waiting
PROMPT
Oct 23 23:39:12.35|0x7f82daebb700|0_1_1540337949.313|5001|VXIpromptWait|EVT
< 1 : prompt|session=1|item=1|result=ok
Oct 23
23:39:12.35|0x7f82daebb700|0_1_1540337949.313|8002|fr.ulex.vxi|VXI::DoInnerJ
ump()
Oct 23
23:39:12.35|0x7f82daebb700|0_1_1540337949.313|4000|SBjsiCheckVar|entering:
0x7f82d412af00, '$_internalName_31627'
Oct 23
23:39:12.35|0x7f82daebb700|0_1_1540337949.313|4004|JsiContext::CheckVar|Chec
k variable $_internalName_31627, context 0x7f82d41656e0
Oct 23 23:39:12.35|0x7f82daebb700|-1|4002||AccessBegin Lock
Oct 23 23:39:12.35|0x7f82daebb700|-1|4002||AccessBegin Locked
Oct 23 23:39:12.35|0x7f82daebb700|-1|4002||AccessEnd Unlock
Oct 23
23:39:12.35|0x7f82daebb700|0_1_1540337949.313|4000|SBjsiCheckVar|exiting:
returned 0
Oct 23
23:39:12.35|0x7f82daebb700|0_1_1540337949.313|4000|SBjsiCheckVar|entering:
0x7f82d412af00, '$_internalName_31628'
Oct 23
23:39:12.35|0x7f82daebb700|0_1_1540337949.313|4004|JsiContext::CheckVar|Chec
k variable $_internalName_31628, context 0x7f82d41656e0
Oct 23 23:39:12.35|0x7f82daebb700|-1|4002||AccessBegin Lock
...
```

The default log bases (used to define tagIDs) :

```
client.cache.diagLogBase                    VXIInteger  2000
client.inet.diagLogBase                     VXIInteger  3000
client.jsi.diagLogBase                      VXIInteger  4000
client.prompt.diagLogBase                   VXIInteger  5000
client.rec.diagLogBase                      VXIInteger  6000
client.tel.diagLogBase                      VXIInteger  7000
client.vxi.diagLogBase                      VXIInteger  8000
client.object.diagLogBase                   VXIInteger  9000
client.client.diagLogBase                   VXIInteger  10000
```

#### Interet connector

```
<DiagnosticMessages moduleName="*SBinet">
    <diag tag="0">SBinet: API trace </diag>
    <diag tag="1">SBinet: Channel diagnostics </diag>
    <diag tag="2">SBinet: Stream diagnostics </diag>
    <diag tag="3">SBinet: Cookie diagnostics </diag>
    <diag tag="4">SBinet: Validator diagnostics </diag>
```

```
    <diag tag="5">SBinet: Cache diagnostics </diag>
    <diag tag="6">SBinet: Timing diagnostics </diag>
    <diag tag="10">SBinet: Dump HTTP requests and responses </diag>

</DiagnosticMessages>
```

### Cache

```
<DiagnosticMessages moduleName="*SBcache">
    <diag tag="0">SBcache: API trace </diag>
    <diag tag="1">SBcache: Cache manager diagnostics </diag>
    <diag tag="2">SBcache: Cache entry diagnostics </diag>
    <diag tag="3">SBcache: Cache stream diagnostics </diag>
    <diag tag="4">SBcache: Cache entry table mutex diagnostics </diag>
    <diag tag="5">SBcache: Cache entry mutex diagnostics </diag>
</DiagnosticMessages>
```

### EcmaScript interpreter

```
<DiagnosticMessages moduleName="*SBjsi">
    <diag tag="0">SBjsi: API trace </diag>
    <diag tag="1">SBjsi: JavaScript context diagnostics </diag>
    <diag tag="2">SBjsi: JavaScript garbage collection trace </diag>
    <diag tag="4">SBjsi: JavaScript scope diagnostics </diag>
    <diag tag="200">SBjsi: Native ScriptEase error messages </diag>
    <diag tag="201">SBjsi: ScriptEase debug log messages </diag>
</DiagnosticMessages>
```

### Client main

```
<DiagnosticMessages moduleName="*OSBclient">
    <diag tag="0">OSBclient: API trace </diag>
    <diag tag="1">OSBclient: Component names and versions </diag>
    <diag tag="2">OSBclient: General diagnostics </diag>
</DiagnosticMessages>
```

### Objects

```
<DiagnosticMessages moduleName="*OSBobject">
    <diag tag="0">OSBobject: API trace </diag>
</DiagnosticMessages>
```

### VoiceXML interpreter (VXI)

```
<DiagnosticMessages moduleName="*.vxi">
    <diag tag="0">VXI: VoiceXML document and application warnings</diag>
    <diag tag="1">VXI: VoiceXML log element output</diag>
    <diag tag="2">VXI: VoiceXML element logging</diag>
    <diag tag="3">VXI: VoiceXML grammar logging</diag>
    <diag tag="4">VXI: VoiceXML transitions</diag>
    <diag tag="5">VXI: VoiceXML log Contents (VoiceXML and Data
XML/Json)</diag>
</DiagnosticMessages>
```

### Telephony

```
<DiagnosticMessages moduleName="*.VXItel">
    <diag tag="0">VXItel: Signaling trace </diag>
</DiagnosticMessages>
```

### Prompt

```
<DiagnosticMessages moduleName="*.VXIprompt">
    <diag tag="0">VXIprompt: Prompting trace </diag>
</DiagnosticMessages>
```

### Recognize

```
<DiagnosticMessages moduleName="*.VXIrec">
    <diag tag="0">VXIrec: Recognition trace </diag>
    <diag tag="1">VXIrec: Grammar trace </diag>
</DiagnosticMessages>
```

## Set the right TimeZone

Install the NTP package to synchronize your server to the world's time.

```
#timedatectl set-timezone America/New_York
```

or, for french servers,

```
#timedatectl set-timezone Europe/Paris
```

or automatically,

```
#dpkg-reconfigure tzdata
```

## Monitor the file descriptors

List the file descriptors used by a process :

```
# ls -l /proc/[PID]/fd
```

## Generate a CallStacks or Coredumps

Generate the threads stacks or a coredump file for the process (voximald) :

```
gdb -ex "thread apply all bt" --batch /usr/sbin/voximald $(pidof voximald) >
/var/log/voximal/backtrace-voximald$(date +%s).txt
gdb /usr/sbin/voximald $(pidof voximald) -ex "gcore /tmp/toto.core" --batch
```

## Execute Asterisk with GDB (debugger)

Launch the Asterisk process with the debugger (gdb) :

```
gdb -ex=r --args asterisk -cvvvvvv -U asterisk -G asterisk
```