

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	`-
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	`-
30788	www-data	20	0	350M	6752	1384	S	0.0	0.2	0:00.00	`-
30787	www-data	20	0	350M	6752	1384	S	0.0	0.2	0:00.00	`-
30786	www-data	20	0	350M	6752	1384	S	0.0	0.2	0:00.00	`-
30785	www-data	20	0	350M	6752	1384	S	0.0	0.2	0:00.00	`-
30784	www-data	20	0	350M	6752	1384	S	0.0	0.2	0:00.00	`-
263	mysql	28	8	670M	38608	1668	S	0.0	0.9	1:55.94	`-
2808	mysql	20	0	670M	38608	1668	S	0.0	0.9	0:00.00	`-

F1Help F2Setup F3SearchF4FilterF5SortedF6CollapF7Nice -F8Nice +F9Kill
F10Quit

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 default, 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=08facd13i9e0q9o1m4vav8ghvn8tbagn21bfc5)
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>
```

From:
<https://wiki.voximal.com/> - **Voximal documentation**

Permanent link:
https://wiki.voximal.com/doku.php?id=installation_guide:debug:start&rev=1592400229

Last update: **2020/06/17 13:23**

