

# Nebuloid Type Reference

Copyright ©2002-2003, Astra Network Inc. All Rights Reserved

January 3, 2003

This is documentation for a BETA version of HAFTA. Please be advised that you may encounter problems; please report them so that they may be fixed. The latest information about HAFTA is always available at <http://www.astranetwork.com/hafta/>.

## Contents

<b>1</b>	<b>Nebuloids Types</b>	<b>2</b>
1.1	NULL . . . . .	2
1.2	ALIVE . . . . .	2
1.3	CPU . . . . .	2
1.4	FD . . . . .	2
1.5	GETPIDBYNAME . . . . .	3
1.6	KILL . . . . .	3
1.7	LOG . . . . .	3
1.8	MAXMEM . . . . .	4
1.9	MEM . . . . .	4
1.10	VAR . . . . .	4
1.11	EXIT . . . . .	5
1.12	SH . . . . .	5
1.13	EXEC . . . . .	5

## 1 Nebuloids Types

### 1.1 NULL

<b>name</b>	NULL
<b>module</b>	mod_null
<b>platforms</b>	posix
<b>inst() args</b>	none
<b>get function</b>	Always returns 0.
<b>get() args</b>	none
<b>set function</b>	nothing.

### 1.2 ALIVE

<b>name</b>	ALIVE
<b>module</b>	mod_alive
<b>platforms</b>	posix
<b>inst() args</b>	none
<b>get function</b>	Returns a 0 to signify that the process does not exist and non-zero to signify that it does.
<b>get() args</b>	none
<b>set function</b>	nothing.

### 1.3 CPU

<b>name</b>	CPU
<b>module</b>	mod_cpu
<b>platforms</b>	nto
<b>inst() args</b>	none
<b>get function</b>	Return that percentage of cpu usage used by the process (or whole system, if pid is 0) multiplied by 1000.
<b>get() args</b>	none
<b>set function</b>	nothing.

### 1.4 FD

<b>name</b>	FD
<b>module</b>	mod_fd
<b>platforms</b>	nto
<b>inst() args</b>	none
<b>get function</b>	Returns number of file descriptors used by the process(or whole system, if pid is 0. <i>not done yet</i> )
<b>get() args</b>	none
<b>set function</b>	nothing.

## 1.5 GETPIDBYNAME

<b>name</b>	GETPIDBYNAME
<b>module</b>	mod_getpidbyname
<b>platforms</b>	nto
<b>inst() args</b>	<b>kString</b> String containing name of process.
<b>get function</b>	Returns that pid of the first process that matches the name of the string given on instantiation. 0 if none exist.
<b>get() args</b>	none
<b>set function</b>	nothing.

## 1.6 KILL

<b>name</b>	KILL
<b>module</b>	mod_kill
<b>platforms</b>	posix
<b>inst() args</b>	<b>kInt</b> Integer representing the signal number
<b>get function</b>	Sends signal number to the process. Returns errno from signal().
<b>get() args</b>	none
<b>set function</b>	nothing.

*Should the signal number be give on get instead of on instantiation?*

## 1.7 LOG

<b>name</b>	LOG
<b>module</b>	mod_log
<b>platforms</b>	posix
<b>inst() args</b>	<b>kInt</b> Type of error logging. 0 - stderr, 1 - file, 2 - syslog. <b>kString</b> Log filename, or syslog server address.
<b>get function</b>	Records a log message.
<b>get() args</b>	<b>kInt</b> Severity of log message 0-10 (fatal-debug). <b>kString</b> Log message itself.
<b>set function</b>	nothing

## 1.8 MAXMEM

<b>name</b>	MAXMEM
<b>module</b>	mod_maxmem
<b>platforms</b>	nto
<b>inst() args</b>	none
<b>get function</b>	Returns the total amount of memory in the system(in bytes.)
<b>get() args</b>	none
<b>set function</b>	nothing.

## 1.9 MEM

<b>name</b>	MEM
<b>module</b>	mod_mem
<b>platforms</b>	nto
<b>inst() args</b>	<b>kInt 0</b> just heap <b>1</b> heap/stack <b>2</b> code/data/heap/stack <b>3</b> code/data/heap/stack/shared_objects <b>4</b> code/data/heap/stack/shared_objects/other(memmapped files, shared memory, etc).
<b>get function</b>	Return the amount of memory in bytes used by the process or system.
<b>get() args</b>	none
<b>set function</b>	nothing.

## 1.10 VAR

<b>name</b>	VAR
<b>module</b>	mod_var
<b>platforms</b>	posix
<b>inst() args</b>	<b>kInt</b> Value to be stored in variable.
<b>get function</b>	Return value stored in variable.
<b>get() args</b>	none
<b>set function</b>	Sets value of variable.

### 1.11 EXIT

<b>name</b>	EXIT
<b>module</b>	mod_exit
<b>platforms</b>	posix
<b>inst() args</b>	none.
<b>get function</b>	Used for testing. Immediately exit()'s the program.
<b>get() args</b>	<b>kInt</b> Exit code of the process.
<b>set function</b>	nothing.

### 1.12 SH

<b>name</b>	SH
<b>module</b>	mod_sh
<b>platforms</b>	posix
<b>inst() args</b>	<b>kString</b> Path of your shell.
<b>get function</b>	Starts a shell with the argument given. Returns the pid of the shell or if the argument begins with 'exec' the pid of the program exec'd.
<b>get() args</b>	<b>kString</b> Command line to be executed by the shell.
<b>set function</b>	nothing.
<b>note</b>	This will not work with QNX Neutrino's uesh.

### 1.13 EXEC

<b>name</b>	EXEC
<b>module</b>	mod_exec
<b>platforms</b>	posix
<b>inst() args</b>	none
<b>get function</b>	Fork()'s and exec()'s the program. No arguments are allowed. This is a very special purpose nebuloid and is rarely useful, but it does allow you to execute programs without a shell. Returns the pid of the process.
<b>get() args</b>	<b>kString</b> Program to be executed.
<b>set function</b>	nothing.