

CubeMOM

Command Guide

Copyright

Copyright©2023 LogicCube Inc. All Rights Reserved.

This document must be created, used, or copied only under the license agreement of Logic Cube Inc. All or part of this document can not be copied, reproduced or translated in any way such as electronically, mechanically or manually without permission from Logic Cube Inc.

Document version	date	Software version
1.4	07/2025	CubeMOM Release 2.2.30

Contents

Copyright	2
Contents	3
CubeMOM Command Guide.....	5
add	6
alter	8
assume	10
batch	11
cmstart	12
cmstop	13
del	14
deliver	16
group	17
help	18
info	19
msgview	21
passwd	22
peer	24
quit	25
resetstat	26
smusage	27
start	28
statis	29
status	31
stop	33
trace	34
user	35
version	36
Object Attributes	37

Global Attributes	38
Host Attributes	43
Node Attributes	44
Group Attributes.....	46
Service Attributes.....	48
Address Attributes.....	51

CubeMOM Command Guide

This document describes the CubeMOM command usage and object attributes. Commands are listed in alphabetical order for ease of finding.

The meta characters in this document are listed in the table below.

Character	Description
<>	Mandatory
[]	Optional
	Exclusive selection delimiter among multiple specified values
*	Zero or more characters
?	One character
-	Number range
.	Separate objects
,	Separate items
...	Repeat item zero or more times

add

Syntax

```
add <-a | -b> node.group[.service] (attribute=value, ...) [-as node.group[.service]]
add <-n | -h> <node | host> (attribute=value, ...) [-as <node | host>]
add -addr node.address (attribute=value, ...) [-as node.address]
add <-argu | -para> node (name=value, ...)
```

Remarks

Add a object to configuration file and (shared) memory. Mandatory attributes must be entered, and optional attributes are assigned default values if omitted. See the [Object Attributes](#) table for whether attributes are mandatory and for valid ranges of attribute values.

If a reference object (-as option) of the same type as the object to be added is specified, it is added as the attribute values of the reference object. The attribute values you enter take precedence over the attribute values of the reference object.

- ※ A process group must contain at least one service.
- ※ A service in an adaptor group must specify a valid address name.

Options

Options	Description
-a	Adaptor
-b	Business
-n	Node
-h	Host
-addr	Address
-argu	Argument
-para	Parameter
-as	Reference object

See also

[info](#), [alter](#), [del](#)

Example

Group	add -b ND01.BG01 (exe_name="test1")
	add -b ND02.BG02 (exe_name="test2") -as ND01.BG01
Service	add -b ND01.BG01.SV01 (msg_dst=".AG01")

	add -b ND01.BG01.SV02 -as ND01.BG01.SV01
Address	add -addr ND01.AD01 (client_server="SERVER", port=1212, lcl_addr="MYHOST")
	add -addr ND01.AD02 -as ND01.AD01
Argument	add -argu ND01 (arg01="arg_01", arg02="arg_02")
Parameter	add -para ND01 (par01="par_00", par02="par_02")
Node	add -n ND03 -as ND02
Host	add -h HT02 (host_addr="HOST02")

alter

Syntax

```
alter [-a | -b] node.group[.service] (attribute=value, ...)
alter <-n | -h> <node | host> (attribute=value, ...)
alter -addr node.address (attribute=value, ...)
alter -glob host (attribute=value, ...)
alter <-argu | -para> node (name=value, ...)
```

Remarks

Alter object attribute values in the configuration file. Real-time reflection attributes are reflected in (shared) memory immediately upon command, so no process restart is required. If it is not a real-time reflection attribute, it is reflected when the process of the altered object is restarted. For real-time reflection attributes, see the object attributes table.

※ Multiple objects can be specified using wildcard characters ('*', '?').

Options

Options	Description
-a	Adaptor
-b	Business
-n	Node
-h	Host
-addr	Address
-glob	Global
-argu	Argument
-para	Parameter

See also

[info](#), [add](#), [del](#)

Example

Group	alter ND01.BG* (prc_min_cnt=3, prc_max_cnt=5)
Service	alter ND01.AG*.SV* (min_conn_cnt=1, max_conn_cnt=5)
Address	alter -addr ND01.AD* (conn_try_cnt=3, conn_try_int="3, 2")
Argument	alter -argu ND01 (arg*="argument vaule")
Parameter	alter -para ND01 (par*="parameter value")
Node	alter ND01 (log_que_cnt=20000, log_alt_cnt=2000, log_alt_int=2)
Host	alter -h HT01 (host_desc="host description")

Global alter -glob HT01 (log_alt_cnt=3000, log_alt_int=3)

assume

Syntax

```
assume [node[.group[.process[.service[.connection]]]]]
```

Remarks

Specifies or releases the command target object. If an object is omitted when commanding, the command is assumed to be for the object specified with the “assume” command. Available for status/status/resetstatus, start/stop, and peer/smusage commands.

※ Multiple objects can be specified using wildcard characters (*, ?).

Options

Options	Description
---------	-------------

See also

Example

```
assume  
assume ND01.BG01  
assume *.BG*
```

batch

Syntax

```
batch pathname [-repeat count] [-interval millisecond]
```

Remarks

Commands specified in the file are sequentially executed line by line. “pathname” is the file name including the absolute path. Some commands (assume, batch, help, version, quit) are not available.

Type (CTRL + C) to abort.

Options

Options	Description
-repeat	Number of repetitions
-interval	Command execution interval, milliseconds (1/1000)

See also

Example

```
batch /home/cubemom/commands.txt
```

cmstart

Syntax

```
cmstart [-nonode]
```

Remarks

An administrator command to start up the host. If child objects (nodes) belonging to the host are set to start automatically, they will also start when the host starts.

- ※ Administrator commands can be executed only when the interpreter is started in administrator mode (cmi -admin).

Options

Options	Description
-nonode	Node does not start

See also

[cmstop](#)

Example

```
cmstart  
cmstart -nonode
```

cmstop

Syntax

```
cmstop [-y]
```

Remarks

An administrator command to shut down the host. After shutting down all child objects (nodes) running on the current host, writes logs in the log storage (shared memory) to disk, and shuts down. Shutdown may be delayed if there are many logs in the log store (shared memory).

- ※ Administrator commands can be executed only when the interpreter is started in administrator mode (cmi -admin).

Options

Options	Description
-y	Stop confirmation - yes

See also

[cmstart](#)

Example

```
cmstop
```

del

Syntax

```
del [-a | -b] [-y] node.group[.service]
del <-n | -h> [-y] <node | host>
del -addr [-y] node.address
del <-argu | -para> [-y] node (name, ...)
```

Remarks

Delete objects from configuration file and (shared) memory. If the process of the specified object is running, it is removed from the configuration file and not from (shared) memory. Objects of the running process are removed from (shared) memory when the process stops. Addresses, arguments and parameters are removed regardless of whether the process is started or not.

※ Multiple objects can be specified using wildcard characters ('*', '?').

Options

Options	Description
-a	Adaptor
-b	Business
-n	Node
-h	Host
-addr	Address
-argu	Argument
-para	Parameter
-y	Delete confirmation - yes

See also

[info](#), [add](#), [alter](#)

Example

Group	del -b ND01.BG01 -y
Service	del -b ND01.BG01.SV01 -y
Address	del -addr ND01.AD01 -y
Argument	del -argu ND01 (arg01) -y
Parameter	del -para ND01 (par01) -y
Node	del -n ND03 -y
Host	del -h HT03 -y

deliver

Syntax

```
deliver [-hexa] node.group[.process][.service[.connection]] /string/
```

Remarks

Deliver the string as a command message to the business process. The business process must determine the purpose of the string. The default handling of a received string is to ignore it. The forward slash (/) character delimits the start and end of a string. The string cannot contain forward slash characters.

※ Multiple objects can be specified using wildcard characters (*, ?).

Options

Options	Description
-hexa	Hexadecimal string - Transforms a hexadecimal string into a binary value and deliver it.

Example

```
deliver ND01.BG01 /0123456789/
deliver ND01.BG* /0123456789/
deliver ND01.BG01 -hexa /30313233343536373839/
```

group

Syntax

```
group <-A | -D | -M> [<-u|-ua|-ud> login [,login]]... [-c comment] user_group
group <-L | -U> user_group
```

Remarks

Add/Delete/Modify/Query user groups.

- ※ When delete/modify/query user group, multiple objects can be specified using wildcard characters ('*', '?').

Options

Options	Description
-A	Add
-D	Delete
-M	Modify
-L	Group list
-U	Group member list
-u	Modify group members
-ua	Add group members
-ud	Delete group members
-c	Comment
-gid	Group identification number

See also

[user](#), [passwd](#)

Example

Add group	group -A -c "TEST1" UG01
Delete group	group -D UG01
Midify group	group -M -u user01, user02 UG01
Display groups	group -L *
Display members	group -U *

help

Syntax

```
help [command]
help <-glob | -h | -n | -g | -s | -addr>
```

Remarks

Describes how to use commands or attributes of objects.

Options

Options	Description
-glob	Global attributes
-h	Host attributes
-n	Node attributes
-g	Group attributes
-s	Service attributes
-addr	Address attributes

Example

```
help
help -g
```

info

Syntax

```
info [-a | -b] [-m] [-v] node.group[.service] [(attribute, ...)]
info <-n | -h> [-m] [-v] <node | host> [(attribute, ...)]
info -addr [-m] [-v] node.address [(attribute, ...)]
info -glob [-m] [-v] host [(attribute, ...)]
info <-argu | -para> [-m] node (name, ...)
```

Remarks

Displays the attribute values of objects registered in the configuration file. An asterisk (*) character is appended to the attribute value if the value in the configuration file differs from the value in (shared) memory. The fact that attribute values are different means that it has not been reflected in the process. See the attribute table for which objects need to be restarted to reflect the process.

※ Multiple objects can be specified using wildcard characters (*, ?).

Options

Options	Description
-a	Adaptor
-b	Business
-n	Node
-h	Host
-glob	Global
-addr	Address
-argu	Argument
-para	Parameter
-m	Display attribute values of (shared) memory
-v	Display vertically

See also

[add](#), [alter](#), [del](#)

Example

Group	Info ND01.AP01 (*) -v info ND01.AP01 (grp_type, grp_name) info ND01.AP01 (grp*)
Service	info ND01.AP01.SV01 (*) -v
Address	info -addr ND01.AD01 (*) -v

Argument	info -argu NODE01 (*)
Parameter	info -para NODE01 (*)
Node	info -n ND01 (*) -v
Host	info -h HT01 (*) -v
Global	info -glob HT01 (*) -v

msgview

Syntax

```
msgview [-n] filename key
```

Remarks

Decrypts and displays encrypted messages in a file. If the “-n” option is present, it looks for the file in the core process log directory (“clog”), and if the “-n” option is not present, it looks for the file in the business/adapter process log directory (“plog”).

The argument *key* is the value of the key column of the record in the file.

Options

Options	Description
-n	Core-process log directory("clog")

Example

```
msgview -n NODE00_AP000_20250708.msg 14CF000000000000
```

passwd

Syntax

```
passwd [[-d] [-e] [-l | -u] login]
passwd -policy [(attribute=value) [, (attribute=value)]... ]
```

Remarks

Change a user's password or ivew or change the password policy. Only administrators can change other users' passwords and password policy.

Options

Options	Description
-d	Delete user password - log in without a password
-e	Require password change on next login
-l	Login lock setting - Cannot log in
-u	Unlock login - Login available
-policy	Password policy

Policy attributes

Attributes	Description
min_len	Password minimum length
max_repeat	Number of repetitions of the same character allowed
lchar_cnt	Minimum number of lowercase letters
uchar_cnt	minimum number of uppercase letters
nchar_cnt	minimum number of digits
schar_cnt	minimum number of special characters
char_dist	Number of characters different from previous password
deny_cnt	Number of consecutive login failures for auto-lock
deny_unlock_sec	Auto-unlock time (seconds) - Elapsed time since last login failure

See also

[group](#), [user](#)

Example

```
passwd -d user00
passwd -e user00
passwd -policy
passwd -policy min_len=5, max_repeat=3
```


peer

Syntax

```
peer <-a | -b> node.group.process.service.connection
peer <-n | -h> <node | host>.group.process.service.connection
```

Remarks

Displays internal and external internet addresses and port numbers. The core process name (host:CML/CMU, node:CMC/CMW/CMP) corresponds to the group and the process is ignored.

※ Multiple objects can be specified using wildcard characters ('*', '?').

Options

Options	Description
-a	Adaptor
-b	Business
-n	Node (CMC/CMW/CMP)
-h	Host (CML/CMU)

Example

```
peer ND01.AG01.1.SV01.1
peer ND01.AG01.1.SV01.*
peer -n ND01.*.*.*
peer -h HT01.*.*.*
```

quit

Syntax

```
quit
```

Remarks

Quit the command interpreter.

Options

Options	Description
---------	-------------

Example

```
quit
```

resetstatis

Syntax

```
resetstatis [-a | -b] [-s object_state] node[.group[.process[.service[.connection]]]]
resetstatis <-n | -h> [-s object_state] <node | host>[.group[.process[.service[.connection]]]]
```

Remarks

Initialize the statistics of the object (queue, number of sending/receiving, TPS, average response time). The core process name (host:CML/CMU, node:CMC/CMW/CMP) corresponds to the group and the process is ignored.

※ You can specify multiple objects using wildcard characters (*, ?).

Options

Options	Description
-a	Adaptor
-b	Business
-n	Node
-h	Host

See also

[status](#), [statis](#)

Example

Group	resetstatis ND01.AG01 resetstatis ND01.*
Process	resetstatis ND01.AG01.1 resetstatis ND01.AG01.1-2
Service	resetstatis ND01.AG01.1.SV01 resetstatis ND01.AG01.*.*
Task/Connection	resetstatis ND01.AG01.1.SV01.1 resetstatis ND01.AG01.1.SV01.1-2 resetstatis ND01.AG01.1.SV01.*
Core process	resetstatis -n ND01.* resetstatis -h HT01.*

smusage

Syntax

```
smusage <-n | -h> <node | host>
```

Remarks

Displays shared memory usage and utilization.

※ You can specify multiple objects using wildcard characters (*, ?).

Options

Options	Description
-n	Node
-h	Host

Example

Node	smusage -n ND01
Host	smusage -h HT01

start

Syntax

```
start [-a | -b] node.group[.process[.service[.connection]]]
start -n [host.]node
```

Remarks

Start the object. If the child objects are set to start automatically, the child objects are also started. If the object is an unconnected TCP client connection, it tries to connect to the server. If the object is a node and the host name is omitted, it starts on the host belonging to the node. Only administrators can start a node.

※ Multiple objects can be specified using wildcard characters ('*', '?').

Options

Options	Description
-a	Adaptor
-b	Business
-n	Node

See also

[stop](#)

Example

Group	start ND01.AG01 start ND01.*
Process	start ND01.AG01.1 start ND01.AG01.1-2 start ND01.AG01.*
Service	start ND01.AG01.1.SV01 start ND01.AG01.*.*
Connection	start ND01.AG01.1.SV01.1 start ND01.AG01.1.SV01.1-2 start ND01.AG01.1.SV01.*
Node	start -n ND01

statis

Syntax

```
statis [-a | -b] [-s object_state] node[.group[.process[.service[.connection]]]]
statis <-n | -h> [-s object_state] <node | host>[.group[.process[.service[.connection]]]]
```

Remarks

Query the statistics (queue, number of sending/receiving, TPS, average response time) of the object. The core process name (host:CML/CMU, node:CMC/CMW/CMP) corresponds to the group and the process is ignored.

※ Multiple objects can be specified using wildcard characters ('*', '?').

Options

Options	Description
-a	Adaptor
-b	Business
-n	Node (CMC/CMW/CMP)
-h	Host (CML/CMU)
-s	<ul style="list-style-type: none"> Object state Connection : disconnected/connected/disconnecting/connecting Non-Connection : stopped/running/stopping/starting/abnormal

Columns

Columns	Description
NAME	Object name
STAT	Object state
QUEUE-MAX	The maximum number of messages that were queued
QUEUE-FULL	The number of messages that exceeded the maximum size of the queue
RECV	Number of messages received - Core process perspective
SEND	Number of messages sent - Core process perspective
TPS(M)	Current TPS(Transaction Per Second) or TPM(Transaction Per Minute)
AVG-ETIME	Adapter Process - Transaction average elapsed time Business Process - Average message processing time for business process

See also

[status](#), [resetstatis](#)

Example

Group	statis ND01.AG01
	statis ND01.*
Process	statis ND01.AG01.1
	statis ND01.AG01.1-2
Service	statis ND01.AG01.1.SV01
	statis ND01.AG01.*.*
Connection	statis ND01.AG01.1.SV01.1
	statis ND01.AG01.1.SV01.1-2
	statis ND01.AG01.1.SV01.*
Core Process	statis -n ND01.*
	statis -h HT01.*

status

Syntax

```
status [-a | -b] [-s object_state] node[.group[.process[.service[.connection]]]]
status <-n | -h> [-s object_state] <node | host>[.group[.process[.service[.connection]]]]
```

Remarks

Query object status (PID, queue usage, status change time). The core process name (host:CML/CMU, node:CMC/CMW/CMP) corresponds to the group and the process is ignored.

※ Multiple objects can be specified using wildcard characters ('*', '?').

Options

Options	Description
-a	Adaptor
-b	Business
-n	Node (CMC/CMW/CMP)
-h	Host (CML/CMU)
-s	<ul style="list-style-type: none"> Object state Connection : disconnected/connected/disconnecting/connecting Non-Connection : stopped/running/stopping/starting/abnormal

Columns

Columns	Description
NAME	Object name
STAT	Object state
PID	PID of the process (which contains the object)
QUEUE	Queue usage
SDATE	Date and time of object state change

See also

[statis](#), [resetstatis](#)

Example

Group	status ND01.AG01 status ND01.*
Process	status ND01.AG01.1 status ND01.AG01.1-2
Service	status ND01.AG01.1.SV01

	status ND01.AG01.*.*
Connection	status ND01.AG01.1.SV01.1
	status ND01.AG01.1.SV01.1-2
	status ND01.AG01.1.SV01.*
Core Process	status -n ND01.*
	status -h HT01.*

stop

Syntax

```
stop [-a | -b] node.group[.process[.service[.connection]]]
stop -n [-y] Node
```

Remarks

Stop the object. Stop after stopping all child objects. If the object is a connected connection, the connection is disconnected. If the object is a node, all logs in the log storage are written to disk and then stopped. Stopping may be delayed if there are many logs in the log store. Nodes can only be stopped by administrators.

※ Multiple objects can be specified using wildcard characters ('*', '?').

Options

Options	Description
-a	Adaptor
-b	Business
-n	Node
-y	Stop confirmation - yes

See also

[start](#)

Example

Group	stop ND01.AG01
	stop ND01.*
Process	stop ND01.AG01.1
	stop ND01.AG01.1-2 stop ND01.AG01.*
Service	stop ND01.AG01.1.SV01
	stop ND01.AG01.*.*
Connection	stop ND01.AG01.1.SV01.1
	stop ND01.AG01.1.SV01.1-2
	stop ND01.AG01.1.SV01.*

trace

Syntax

```
trace [-d date] XID
```

Remarks

Trace message flow and process elapsed time. If the date is omitted, it will be queried as of today. Message tracing is performed through a separate log and is generated when the trace log level is enabled. The message tracking identifier (XID) can be obtained from the log file.

Options

Options	Description
-d	Date (in the form of global attribute "date_format")

Columns

Columns	Description
PROCESS	Process sent or received
SR	Sending or Receiving - Core process perspective
TIME	Sending or Receiving time - Core process time
MTYPE	Message type
XID	Message tracking identifier
SEQ	Message tracking identifier, sequence number
LEN	Message length
SRC	Message source
DST	Message destination
ERROR	Error message

Example

```
trace 0100200100000063
```

user

Syntax

```
user <-A | -D | -M> [-g user_group] [-og user_group [, user_group]...] [-p password] [-e expire] [-f
inactive] login
user <-L | -G> login
```

Remarks

Add/Delete/Modify/Query user.

- ※ When delete/modify/query user, multiple objects can be specified using wildcard characters ('*', '?').

Options

Options	Description
-A	Add
-D	Delete
-M	Modify
-L	User list
-G	User group
-g	User's primary group
-og	User's additional groups
-c	User comment
-p	Password
-e	Account expiry date (yyyymmdd)
-f	Password expiration date - Number of days
-uid	User identification number

See also

[group](#), [passwd](#)

Example

Add user	user -A -g UG01 -og UG02, UG03 user01
Delete user	user -D user01
Midify user	user -M -g UG02 -og UG03, UG04
Display user list	user -L *
Display user group	user -G *

version

Syntax

```
version
```

Remarks

Display CubeMOM version.

Options

Options	Description
---------	-------------

Example

```
version
```

Object Attributes

Describes all attributes of an object. The columns of the attribute table are shown in the following table.

Columns	Description
ATTRIBUTE	Attribute name
TYPE	Attribute type STR : String INT : Integer HEX : Hexadecimal CODE : Code
RS	Upon reflection of changes, restart object H : Host N : Node G : Group P : Process S : Service X : Not required
MIN	Integer : minimum value. String : minimum length Code : minimum value
MAX	Integer: maximum value String : max length Code : maximum value
DEFAULT	Integer : default value
IN	Mandatory : Y/N

※ If the attribute type is code, when adding or modifying an object, you must enter a code value.

Global Attributes

ATTRIBUTE	TYPE	RS	MIN	MAX	DEFAULT	IN
adpt_exe_path	STR	H	1	255	N/A	N
	Adapter process, executable file full path. ※ You must add the path to the Shell "PATH" environment variable.					
busi_exe_path	STR	H	1	255	N/A	N
	Business process, executable file full path. ※ You must add the path to the Shell "PATH" environment variable.					
log_storage_size	INT	H	1000	3000000	100000	N
	Host log store (queue) size (number of records). ➤ Host process (CML, CMU) logs ➤ Trace log					
log_alt_size	INT	H	800	3000000	80000	N
	Host log storage (queue) alarm water mark (number of records).					
log_alt_int	INT	H	0	3600	1	N
	Host log storage (queue) alarm interval (seconds). ➤ 0 : No alarm					
log_srcdst_len	INT	H	15	87	50	N
	Number of source and destination column bytes in the log.					
log_file_mode	CODE	H	100	777	644	N
	Log file access permissions for Owner, Group, and Others.					
	➤ 'r' : Read permission					
	➤ 'w' : Write permission					
	➤ 'x' : Execute permission					
➤ '-' : No permission						
Ex) alter -glob HT01 (log_file_mode = "rw-r--r--")						
log_format	CODE	H	1	12	N/A	N
	Log format (column order) - Column can be omitted.					
	➤ "level" : Log level					
	➤ "logc" : Log classification code					
	➤ "time" : Time					
	➤ "thrd" : Thread number					
	➤ "type" : Message type					
	➤ "xid" : Message trace ID					
	➤ "seq" : Message trace ID sequence					
	➤ "src" : Message source					
➤ "dst" : Message destination						
➤ "rot" : Message routing destination						
➤ "len" : Message length						
➤ "text" : Error message or description						
Ex) alter -glob HT01 (log_format = "level logc time thrd type xid seq src dst rot len text")						
clog_path	STR	H	1	255	N/A	N
	Core process log full path.					
plog_path	STR	H	1	255	N/A	N
	Adapter and business process log full path.					
cml_thd_cnt	INT	X	2	512	16	N
	The number of threads in the CML process.					
cmu_thd_cnt	INT	X	2	512	8	N
	The number of threads in the CMU process.					
brdc_xid_inc	INT	X	1	2147483647	100	N
	Broadcast message, XID sequence increment value.					

	CODE	X	1	2147483647	7	N
cml_loglevel	CML process loglevel. One of the essential options must be specified.					
	<ul style="list-style-type: none"> ➤ "critical" : Essential option ➤ "error" : Essential option ➤ "warning" : Essential option ➤ "info" : Essential option ➤ "verbose" : Essential option ➤ "debug" : Essential option ➤ "trace" : Additional option 					
	Ex) alter -glob HT01 (cml_loglevel = "debug+trace")					
cmu_loglevel	CODE	X	1	2147483647	7	N
	CMU process loglevel. The code value is the same as "cml_loglevel".					
cml_conn_cnt	INT	H	1	10	5	N
	The number of connections that the CML process will connect to other hosts (CML).					
cmu_conn_cnt	INT	H	1	10	5	N
	The number of connections that the CMU process will connect to the same host (CML).					
cmc_conn_cnt	INT	H	1	10	5	N
	The number of connections the CMC process will connect to the host (CML).					
cmp_conn_cnt	INT	H	1	100	10	N
	The number of connections that the CMP process will connect to other nodes (CMP).					
cmx_idle_tmout	INT	H	0	3600	0	N
	Core process (CMX) connection idle timeout (seconds).					
	➤ 0 : No timeout					
cmx_rcv_tmout	INT	H	0	3600	1	N
	Core process (CMX) connection receive timeout (seconds).					
	➤ 0 : No timeout					
cmx_send_tmout	INT	H	0	3600	1	N
	Core process (CMX) connection send timeout (seconds).					
	➤ 0 : No timeout.					
cmx_que_tmout	INT	H	0	3600	1	N
	Core process (CMX), queued message (processing) wait timeout (seconds).					
	➤ 0 : No timeout					
host_shm_key	STR	H	1	15	N/A	N
	Host shared memory key (hexadecimal string).					
host_sem_key	STR	H	1	15	N/A	N
	Host semaphore key (hexadecimal string).					
logq_sem_key	STR	H	1	15	N/A	N
	Host log semaphore key (hexadecimal string).					
hstf_sem_key	STR	H	1	15	N/A	N
	Host configuration file semaphore key (hexadecimal string).					
usrf_sem_key	STR	H	1	15	N/A	N
	User configuration file semaphore key (hexadecimal string).					
cml_que_size	INT	H	10	10000	300	N
	CML, CMU process IPC queue size.					
cml_socket_domain	CODE	H	2	3	2	N
	CML process, form of socket address.					
	<ul style="list-style-type: none"> ➤ "IPv4" : Internet protocol version 4 ➤ "IPv6" : Internet protocol version 6 					
cml_ipc_port	INT	H	1	65535	3377	N
	CML process, IPC port number.					
cml_cmi_port	INT	H	1	65535	3378	N

	CML process , CMI(Command-line Interface) port number.					
cml_gui_port	INT	H	1	65535	3379	N
	CML process , GUI(Graphical User Interface) port number.					
	CODE	H	2	3	2	N
cmp_socket_domain	CMP process, form of socket address.					
	➤ "IPv4" : Internet protocol version 4					
	➤ "IPv6" : Internet protocol version 6					
cmp_port	INT	H	1	65535	9191	N
	CMP process, port number.					
cml_so_keepalive	INT	H	0	1	0	N
	CML process, socket option (SO_KEEPALIVE).					
cml_so_linger	INT	H	0	2147483647	0	N
	CML process, socket option (SO_SO_LINGER).					
cml_so_rcvbuf_size	INT	H	0	2147483647	32768	N
	CML process, socket option (SO_RCVBUF).					
cml_so_sendbuf_size	INT	H	0	2147483647	32768	N
	CML process, socket option (SO_SO_SNDBUF).					
cml_so_reuseaddr	INT	H	0	1	0	N
	CML process, socket option (SO_REUSEADDR).					
cml_so_reuseport	INT	H	0	1	0	N
	CML process, socket option (SO_REUSEPORT).					
cml_tcp_nodelay	INT	H	0	1	0	N
	CML process, socket option (TCP_NODELAY).					
cml_tcp_keepidle	INT	H	0	65000	0	N
	CML process, Socket option(TCP_KEEPIDLE).					
cml_tcp_keepintvl	INT	H	0	250	0	N
	CML process, Socket option(TCP_KEEPINTVL).					
cml_tcp_keepcnt	INT	H	0	250	0	N
	CML process, Socket option(TCP_KEEPCNT).					
cmu_so_keepalive	INT	H	0	1	0	N
	CMU process, socket option (SO_KEEPALIVE).					
cmu_so_linger	INT	H	0	2147483647	0	N
	CMU process, socket option (SO_SO_LINGER).					
cmu_so_rcvbuf_size	INT	H	0	2147483647	32768	N
	CMU process, socket option (SO_RCVBUF).					
cmu_so_sendbuf_size	INT	H	0	2147483647	32768	N
	CMU process, socket option (SO_SO_SNDBUF).					
cmu_so_reuseaddr	INT	H	0	1	0	N
	CMU process, socket option (SO_REUSEADDR).					
cmu_so_reuseport	INT	H	0	1	0	N
	CMU process, socket option (SO_REUSEPORT).					
cmu_tcp_nodelay	INT	H	0	1	0	N
	CMU process, socket option (TCP_NODELAY).					
cmu_tcp_keepidle	INT	H	0	65000	0	N
	CMU process, socket option (TCP_KEEPIDLE).					
cmu_tcp_keepintvl	INT	H	0	250	0	N
	CMU process, socket option (TCP_KEEPINTVL).					
cmu_tcp_keepcnt	INT	H	0	250	0	N
	CMU process, socket option (TCP_KEEPCNT).					
cmc_so_keepalive	INT	H	0	1	0	N
	CMC process, socket option (SO_KEEPALIVE).					
cmc_so_linger	INT	H	0	2147483647	0	N
	CMC process, socket option (SO_SO_LINGER).					
cmc_so_rcvbuf_size	INT	H	0	2147483647	32768	N
	CMC process, socket option (SO_RCVBUF).					
cmc_so_sendbuf_size	INT	H	0	2147483647	32768	N
	CMC process, socket option (SO_SO_SNDBUF).					

cmc_so_reuseaddr	INT	H	0	1	0	N
	CMC process, socket option (SO_REUSEADDR).					
cmc_so_reuseport	INT	H	0	1	0	N
	CMC process, socket option (SO_REUSEPORT).					
cmc_tcp_nodelay	INT	H	0	1	0	N
	CMC process, socket option (TCP_NODELAY).					
cmc_tcp_keepidle	INT	H	0	65000	0	N
	CMC process, socket option (TCP_KEEPIDLE).					
cmc_tcp_keepintvl	INT	H	0	250	0	N
	CMC process, socket option (TCP_KEEPINTVL).					
cmc_tcp_keepcnt	INT	H	0	250	0	N
	CMC process, socket option (TCP_KEEPCNT).					
cmp_so_keepalive	INT	H	0	1	0	N
	CMP process, socket option (SO_KEEPALIVE).					
cmp_so_linger	INT	H	0	2147483647	0	N
	CMP process, socket option (SO_SO_LINGER).					
cmp_so_rcvbuf_size	INT	H	0	2147483647	32768	N
	CMP process, socket option (SO_RCVBUF).					
cmp_so_sendbuf_size	INT	H	0	2147483647	32768	N
	CMP process, socket option (SO_SO_SNDBUF).					
cmp_so_reuseaddr	INT	H	0	1	0	N
	CMP process, socket option (SO_REUSEADDR).					
cmp_so_reuseport	INT	H	0	1	0	N
	CMP process, socket option (SO_REUSEPORT).					
cmp_tcp_nodelay	INT	H	0	1	0	N
	CMP process, socket option (TCP_NODELAY).					
cmp_tcp_keepidle	INT	H	0	65000	0	N
	CMP process, socket option (TCP_KEEPIDLE).					
cmp_tcp_keepintvl	INT	H	0	250	0	N
	CMP process, socket option (TCP_KEEPINTVL).					
cmp_tcp_keepcnt	INT	H	0	250	0	N
	CMP process, socket option (TCP_KEEPCNT).					
ui_max_conn_cnt	INT	H	100	1000	100	N
	GUI and CMI, maximum number of connections. ➤ $ui_max_conn_cnt \geq gui_max_conn_cnt + cmi_max_conn_cnt$					
gui_max_conn_cnt	INT	H	1	1000	50	N
	GUI maximum number of connections.					
cmi_max_conn_cnt	INT	H	1	1000	10	N
	CMI maximum number of connections.					
ui_conn_que_size	INT	H	1	1000	10	N
	GUI and CMI, connection queue size for response to requests.					
ui_idle_tmout	INT	H	0	3600	60	N
	GUI and CMI, connection idle timeout (seconds) - disconnection if timeout. ➤ 0 : No timeout					
ui_rcv_tmout	INT	H	0	3600	5	N
	GUI and CMI, connection receive timeout (seconds) - disconnection if timeout. ➤ 0 : No timeout					
ui_send_tmout	INT	H	0	3600	5	N
	GUI and CMI, connection send timeout (seconds) - disconnection if timeout. ➤ 0 : No timeout					
ui_que_tmout	INT	H	0	3600	10	N
	GUI and CMI, queued message (sending) wait timeout (seconds). ➤ 0 : No timeout					
cmu_cmd_que_size	INT	H	10	10000	100	N
	CMU process, command queue size.					

cmu_cmd_que_tmout	INT	H	1	30	3	N
	CMU process, command (queue) wait timeout (seconds).					
date_format	CODE	H	1	3	1	N
	Date format.					
	➤ "yyyymmdd" : Year, month, date format					
	➤ "mmddyyyy" : Month, date, year format					
	➤ "ddmmyyyy" : Date, month, year format					
date_sepchr	INT	H	0	255	47	N
	Date separator character, ASCII code value .					
	➤ 0 : No separator					
time_sepchr	INT	H	0	255	58	N
	Time separator character, ASCII code value.					
	➤ 0 : No separator					
log_delay_tm	INT	X	0	3600000	1000	N
	Log write delay alarm (1/1000 of a second).					
	➤ 0 : No alarm					

Host Attributes

ATTRIBUTE	TYPE	RS	MIN	MAX	DEFAULT	IN
host_name	STR	H	1	15	N/A	Y
	Host name.					
host_desc	STR	H	1	47	N/A	N
	Host description.					
host_addr	STR	H	1	47	N/A	Y
	Host address (IP/host-name/domain).					

Node Attributes

ATTRIBUTE	TYPE	RS	MIN	MAX	DEFAULT	IN
node_name	STR	H	1	15	N/A	Y
	Node name.					
node_desc	STR	X	1	47	N/A	N
	Node description.					
host_name	STR	H	1	15	N/A	Y
	The node's belonging hostname.					
auto_start	INT	X	0	1	1	N
	Whether to start the node when the host starts.					
node_fname	STR	H	1	15	N/A	N
	Node configuration file name.					
backup_node	STR	H	1	15	N/A	N
	Backup node name.					
cmc_loglevel	CODE	X	1	2147483647	7	N
	CMC process loglevel. One of the essential options must be specified.					
	<ul style="list-style-type: none"> ➤ "critical" : Essential option ➤ "error" : Essential option ➤ "warning" : Essential option ➤ "info" : Essential option ➤ "verbose" : Essential option ➤ "debug" : Essential option ➤ "trace" : Additional option 					
cmw_loglevel	CODE	X	1	2147483647	7	N
	CMW process loglevel. The code value is the same as "cmc_loglevel".					
cmp_loglevel	CODE	X	1	2147483647	7	N
	CMP process loglevel. The code value is the same as "cmc_loglevel".					
cmw_conn_cnt	INT	N	1	10	2	N
	Same node CMC process connection, number of CMW connections.					
cmp_conn_cnt	INT	N	1	50	5	N
	Same node CMC process connection, number of CMP connections.					
cmx_que_size	INT	N	10	10000	300	N
	CMC/CMW/CMP process IPC queue size.					
cmp_node_addr	STR	H	1	47	N/A	Y
	CMP process address (IP/host-name/domain).					
node_shm_key	STR	N	1	15	N/A	Y
	Node shared memory key (hexadecimal string).					
node_sem_key	STR	N	1	15	N/A	Y
	Node semaphore key (hexadecimal string).					
logq_sem_key	STR	N	1	15	N/A	Y
	Node log queue, semaphore key (hexadecimal string).					
nodf_sem_key	STR	N	1	15	N/A	Y
	Node configuration file, semaphore key (hexadecimal string).					
perr_sem_key	STR	N	1	15	N/A	Y
	Adapter and business process error logs, semaphore key (hexadecimal string).					
node_start_seq	INT	N	1	50000	100	N
	Node startup sequence number. Smaller number start first.					
node_start_tmout	INT	X	1	3600	30	N
	Node startup timeout (seconds).					
tran_cnt_type	CODE	N	1	2	1	N
	Transaction count type.					
	<ul style="list-style-type: none"> ➤ "TPS" : Transaction per second ➤ "TPM" : Transaction per minute 					

grp_start_int	INT	X	0	1000	0	N
	Group start-up interval (1/1000 of a second) when starting a node.					
	➤ 0 : No group startup interval					
msg_save_drop	INT	X	0	1	0	N
	If the core process must discard a message, whether to discard it after saving it to a file.					
	➤ 0 : Discard without saving to file					
	➤ 1 : Discard after saving to file					
grp_max_cnt	INT	N	10	10000	1000	N
	The maximum number of process groups within a node.					
cmc_thd_cnt	INT	X	1	512	16	N
	The number of threads in the CMC process.					
cmw_thd_cnt	INT	X	1	512	8	N
	The number of threads in the CMW process.					
cmp_thd_cnt	INT	X	1	512	8	N
	The number of threads in the CMP process.					
grp_storage_size	INT	N	1000	3000000	100000	N
	Group storage size (number of records).					
que_storage_size	INT	N	1000	3000000	200000	N
	Queue storage size (number of records).					
msg_storage_size	INT	N	8388608	2147483647	67108864	N
	Message store size (bytes: 2^N).					
msg_min_alloc_size	INT	N	512	1024	512	N
	Minimum message allocation size (bytes: 2^N).					
msg_max_alloc_size	INT	N	1024	524288	65536	N
	Maximum message allocation size (bytes: 2^N).					
	➔ All messages cannot exceed the length minus 'CubeMOM internal header size (less than 512 bytes)' from 'Maximum message allocation size'.					
addr_storage_size	INT	N	10	1000000	3000	N
	Address storage size (number of records).					
argu_storage_size	INT	N	10	1000000	1000	N
	Argument store size (number of records).					
para_storage_size	INT	N	10	1000000	1000	N
	Parameter store size (number of records).					
log_storage_size	INT	N	1000	3890000	100000	N
	Node log store (queue) size (number of records).					
	➤ Node process (CMC, CMW, CMP) logs					
log_alt_size	INT	N	1000	3890000	80000	N
	Node log storage (queue) alarm water mark (number of records).					
log_alt_int	INT	N	0	3600	1	N
	Node log storage (queue) alarm interval (seconds).					
	➤ 0 : No alarm					

Group Attributes

ATTRIBUTE	TYPE	RS	MIN	MAX	DEFAULT	IN
grp_type	CODE	G	1	2	N/A	Y
	Process group type. <ul style="list-style-type: none"> ➤ "AP" : Adaptor ➤ "BP" : Business 					
grp_name	STR	G	1	15	N/A	Y
Process group name.						
grp_desc	STR	X	1	47	N/A	N
Process group description.						
exe_name	STR	P	1	15	N/A	Y
	Process executable file name. For Java programs, specify "jrun" and use the "argu_name" attribute to specify jvm options, class name, and arguments. "jrun" is a cubemom program that loads the jvm. Execution options are the same as java execution options.					
argu_name	STR	P	1	15	N/A	N
	Process argument name. If the starting character is '@', it means an argument file. The rest except the '@' character corresponds to the file name, and the contents of the file are passed as process execution arguments. The argument file must exist in the path of the "busi_exe_path" attribute value.					
prc_log_key	STR	G	1	15	N/A	N
Process log file semaphore key (hexadecimal string).						
auto_start	INT	X	0	1	1	N
Whether to start the group when node starts.						
brdc_perm	CODE	X	0	2	1	N
	Broadcasting permission. <ul style="list-style-type: none"> ➤ "NA" : Not allow ➤ "SN" : Self node only ➤ "ON" : Self and other node 					
noss_que_opt	CODE	X	1	2	1	N
	Queue option if there are no sendable sessions. <ul style="list-style-type: none"> ➤ "EQ" : En-queue ➤ "EH" : Error handling 					
grp_owner	STR	X	1	15	N/A	N
Process group owner.						
grp_perm	CODE	X	0	333	310	N
	Group permission. <ul style="list-style-type: none"> ➤ 'a' : Alter permission ➤ 'x' : Execute permission ➤ '-' : No permission 					
prc_user_deci	INT	X	0	1	0	N
Whether the user starts/stops the process (by command). If this value is 1, the process is not automatically started/stopped by the CMC.						
msg_max_retry	INT	X	0	256	1	N
	The maximum number of times a message can be send to (another) process in the group if the process received the message stops abnormally. <ul style="list-style-type: none"> ➤ 0 : Send to process unless waiting timeout in queue 					
cmc_loglevel	CODE	X	1	2147483647	271	N
	CMC process loglevel. One of the essential options must be specified. <ul style="list-style-type: none"> ➤ "critical" : Essential option ➤ "error" : Essential option ➤ "warning" : Essential option ➤ "info" : Essential option ➤ "verbose" : Essential option ➤ "debug" : Essential option 					

Service Attributes

ATTRIBUTE	TYPE	RS	MIN	MAX	DEFAULT	IN
svc_name	STR	G	1	15	N/A	Y
	Service name.					
svc_desc	STR	X	1	47	N/A	N
	Service description.					
addr_name	STR	G	1	15	N/A	N
	Address name.					
auto_start	INT	G	0	1	1	N
	Whether to start the service when process starts.					
conn_type	CODE	G	1	2	2	N
	Connection type. > "NP" : Stay connected non-persistently > "PE" : Stay connected persistently					
sdrv_attr	CODE	G	1	3	3	N
	Send/Receive attribute. > "RECV" : Receive only > "SEND" : Send only > "SDRV" : Send and receive					
min_conn_cnt	INT	G	0	65535	1	N
	Minimum number of connections.					
max_conn_cnt	INT	G	1	65535	1	N
	Maximum number of connections.					
msg_ctrl_flg	HEX	X	0	2147483647	0	N
	Message control flags. It is a value set in the source, and determines whether to 'forward to the backup node' or 'return to the source' in case of delivery failure to the destination – 'forward to the backup node' takes precedence.					

※ If the backup node fails again, it is determined again according to the type of failure.

• Failure Type

msg_ctrl_flg	DTNE	Destination not exist
	DTNR	Destination not running
	DTNS	Destination have not sendable session
	DTQF	Destination queue full
	QTMO	Queue wait timeout
	CPNR	Core process not running
	CPQF	Core process queue full
	CPQT	Core process queue wait timeout

The bit numbers for each failure type of the 'message control flags' are as follows. If both the 'forward to the backup node' (P) bit and the 'return to the source' (R) bit are OFF, the message is dropped in case of a failure.

• Bit Numbers

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P	R	P	R	P	R	P	R	P	R	-	R	-	R	-	R
DTNE		DTNR		DTNS		DTQF		QTMO		CPNR		CPQF		CPQT	

Ex) alter ND01.BP01.SV01 (msg_ctrl_flg="FFD5")

• Bit Numbers of "FFD5"

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	1	1	1	1	1	1	0	1	0	1	0	1
DTNE		DTNR		DTNS		DTQF		QTMO		CPNR		CPQF		CPQT	

CODE X 1 2147483647 271 N

svc_loglevel

Service loglevel. One of the essential options must be specified.

- "critical" : Essential option
- "error" : Essential option
- "warning" : Essential option
- "info" : Essential option
- "verbose" : Essential option
- "debug" : Essential option
- "trace" : Additional option

STR X 1 87 N/A Y

Message destination.

- ※ General form : [ND].PG[.[PNUM][.SVC[.CNUM[#PID.SNUM]]]]
- Node name omitted : current node

Ex)

msg_dst

.PG	Any task/connection that can be sent within the group
.PG.BAL	Any task/connection that can be sent within the service
.PG.1	Any task/connection that can be sent within the process
.PG.1.BAL	Any task/connection that can be sent within the service of ".PG.1.BAL"
.PG.1.BAL.1	Task/Connection of ".PG.1.BAL.1"
.PG.1.BAL.1#43.3	Session "43.3" of ".PG.1.BAL.1"
.*.*.*	All task/connection
.PG.1-2.BAL.-	All task/connection of ".PG.1-2.BAL"

INT G 0 2147483647 65536 N

msg_max_len

Maximum message length limit value - disconnection when exceeded.

- 0 : No limit.
- ※ All messages cannot exceed the length minus 'CubeMOM internal header size (512 bytes or less)' from 'msg_max_alloc_size'.

recvbuf_size

INT G 1024 1048576 4096 N

Message receive buffer size in bytes.

sendbuf_size

INT G 1024 1048576 4096 N

Message send buffer size in bytes.

idle_tmout

INT G 0 86400 0 N

Idle (no message sent/received) timeout (seconds) - disconnection if timeout.

- 0 : No timeout

recv_tmout

INT G 0 3600 3 N

Message receiving timeout (in seconds) - disconnection if timeout.

- 0 : No timeout

send_tmout

INT G 0 3600 3 N

Message sending timeout (in seconds) - disconnection if timeout.

- 0 : No timeout

recv_limit

INT G 0 2147483647 0 N

Message receive limit count per connection - disconnection if limit is reached.

- 0 : No limit

send_limit	INT	G	0	2147483647	0	N
	Message send limit count per connection - disconnection if limit is reached. ➤ 0 : No limit					
tran_cnt_attr	CODE	G	0	2	0	N
	Attribute for TPS or TPM counting. ➤ "NONE" : No counting ➤ "RECV" : Increase counting when message receive completed ➤ "SEND" : Increase counting when message send completed					
tran_cnt_psum	INT	G	0	1	1	N
	Whether the parent object counts the TPS or TPM.					
elptm_sdrv_attr	CODE	G	0	2	0	N
	Attribute for calculating transaction elapsed time. ➤ "NONE" : No counting ➤ "RECV" : Increase counting when message receive completed ➤ "SEND" : Increase counting when message send completed					
elptm_sdrv_psum	INT	G	0	1	1	N
	Whether the parent object calculating the transaction elapsed time.					
elptm_cache_cnt	INT	G	1	1000000	100	N
	Cache size (count) for calculating transaction elapsed time. ➔ Number of request message XIDs to store until response or timeout					
elptm_cache_timeout	INT	G	1	3600	1	N
	Cache timeout seconds for calculating transaction elapsed time. ➔ Delete request message XID from cache if timeout					
elptm_delay_tm	INT	G	1	60000	500	N
	Transaction time(1/1000 of a second) delay determination time.. ➔ If it exceeds the specified value, a response time delay log is created.					
tls_communication	INT	G	0	1	0	N
	Whether TLS communication or not.					
tls_verify_peer	INT	G	0	1	0	N
	TLS communication, peer authentication or not.					
tls_cert_chain_fn	INT	G	1	15	N/A	N
	TLS communication, certificate (chain) file name.					
tls_pri_key_fn	INT	G	1	15	N/A	N
	TLS communication, private key file name.					
tls_pri_key_pw	INT	G	1	15	N/A	N
	TLS communication, private key password.					
tls_trust_ca_fn	INT	G	1	15	N/A	N
	TLS communication, trusted CA file name.					
tls_cert_common	INT	G	1	31	N/A	N
	TLS communication, common name in certificate.					

Address Attributes

ATTRIBUTE	TYPE	RS	MIN	MAX	DEFAULT	IN
addr_name	STR	G	1	15	N/A	Y
	Address name.					
addr_desc	STR	X	1	47	N/A	N
	Address description.					
socket_domain	CODE	S	2	3	2	N
	Form of socket address .					
	➤ "IPv4" : Internet protocol version 4					
	➤ "IPv6" : Internet protocol version 6					
socket_type	CODE	S	1	1	1	N
	Socket type.					
	➤ "TCP" : Transmission control protocol					
client_server	CODE	S	1	2	1	Y
	Client or Server.					
	➤ "CLIENT" : TCP client					
	➤ "SERVER" : TCP server					
port	INT	S	1	65535	N/A	Y
	Port number.					
lsn_que_size	INT	S	0	4096	0	N
	Server, connection wait queue size (length).					
so_keepalive	INT	S	0	1	0	N
	Socket option (SO_KEEPALIVE).					
so_linger	INT	S	0	2147483647	0	N
	Socket option (SO_SO_LINGER).					
so_rcvbuf_size	INT	S	0	2147483647	32768	N
	Socket option (SO_RCVBUF).					
so_sendbuf_size	INT	S	0	2147483647	32768	N
	Socket option (SO_SO_SNDBUF).					
so_reuseaddr	INT	S	0	1	0	N
	Socket option (SO_REUSEADDR).					
so_reuseport	INT	S	0	1	0	N
	Socket option (SO_REUSEPORT).					
tcp_nodelay	INT	S	0	1	0	N
	Socket option (TCP_NODELAY).					
tcp_keepidle	INT	S	0	65000	0	N
	Socket option (TCP_KEEPIDLE).					
tcp_keepintvl	INT	S	0	250	0	N
	Socket option (TCP_KEEPINTVL).					
tcp_keepcnt	INT	S	0	250	0	N
	Socket option (TCP_KEEPCNT).					
lcl_addr	STR	S	1	47	N/A	N
	Local address (IP/host-name/domain) - TCP Server required.					
rmt_addr	STR	S	1	47	N/A	N
	Remote address (IP/host-name/domain) - TCP Client required.					
msg_delim_type	CODE	S	0	1	1	N
	Message delimiter type.					
	➤ "NONE" : No message delimiter - Message deliver to destination as received from socket.					
	➤ "HEAD" : Delimited message by header					
deliver_header	INT	S	0	1	0	N
	Whether to deliver the message with header to the destination.					
len_header_size	INT	S	0	4096	4	N
	Header size in bytes.					
len_offset	INT	S	0	4096	0	N

			The length field offset in the header.			
	CODE	S	1	6	1	N
len_type	The type of length field in the header.					
	➤ "BIN" : Binary					
	➤ "ANM" : Numeric character(ASCII)					
	➤ "ENM" : Numeric character(EBCDIC)					
	➤ "BCD" : Binary-coded decimal					
	➤ "AHS" : Hexadecimal string(ASCII)					
			➤ "EHS" : Hexadecimal string(EBCDIC)			
len_byte	INT	S	1	16	4	N
	Bytes of the length field in the header.					
len_inc_header	INT	S	0	1	0	N
	Whether the value of the length field in the header includes the header size.					
len_bigendian	INT	S	0	1	1	N
	Whether the byte alignment of the length field in the header is big-endian. Meaningful only when "len_type" is binary.					
conn_try_cnt	INT	S	0	1000000	0	N
	Client, number of connection attempts.					
	➤ 0 : infinity					
conn_try_int	INT	S	0	3600	1	N
	Client, interval(second) between connection attempts.					