

LWCP for VX

Version 0.7.2, Date: 27-Jan-2009

Version 0.7.3, Date: 10-Feb-2009

Version 0.7.4, Date: 16-Feb-2009

Version 0.7.5, Date: 20-Feb-2009 – changed "take studio.next" to "take studio.line"

Version 0.7.6, Date: 02-Mar-2009 – added simple usage example

Version 0.9.6, Date: 02-Sep-2009 – login, mode, date, auto_answer

Sample operations

```
#telnet 192.168.0.24 20518
```

Quick Start

```
# *****
# very basic test to get audio from PC softphone mic to Node
# *****
# logging, default is user = user, password empty
login cc user = "user", password = ""
# *****
# select studio 1
select studio id = 1
# load show 1 in studio 1
select_show studio id = 1
# *****
# now you should call with SIP phone to phone number configured for
#some line. Typically you should be able to call just to IP number
#(192.168.0.24)
# *****
# answer next (first in line) ringing line (next button on desktop director)
take studio.line
# *****
# now there should be audio going from softphones MIC to LW channel
# associated with line 1. For instance 9001.
# *****
# drop line 1. We assume line was ringing as it was configured to
# accept direct IP calls, like this: "sip:192.168.0.24"
drop studio.line#1
# *****
# end basic test
# *****
```

Rest of Commands

```
# login and responses needs to be loget in before sending eny commands
login cc user = "user", password = ""
ack cc logged=FALSE
login cc user = "user", password = ""
ack cc logged=TRUE

# getting studio list
get cc studio_list
indi cc studio_list = [ [1, "Studio 1"] ]

# trying to select non-existing studio
select studio id = 2
ack cc $status="studio : Error during studio selection."

# selecting current studio
select studio id = 1
event studio id = 1, name = "Studio 1", show_id = 1, show_name = "Show 1",
next = 0, num_lines = 12, num_hybrids = 7, num_hyb_fixed = 4, mute = FALSE,
show_locked = FALSE
```

```

get studio id
indi studio id = 1

get studio name
indi studio name = "Studio 1"

get studio show_list
indi studio show_list = [ [1, "Show 1"] ]

# passing unexpected object for function
select_show abc show_id=1
ack cc $status="studio#1: Not a valid object for 'select_show' operation."

# loading show with id 1 in current studio
select_show studio show_id=1
event studio.line#1 state = IDLE, callstate = IDLE, name = null, local =
"<sip:192.168.0.24>", remote = null, fixed = TRUE, hybrid = 1, time = null,
comment = ""
event studio show_id = 1, show_name = "Show 1", next = 0, num_lines = 12,
show_locked = FALSE

# unloading any show from studio
select_show studio show_id=0
event studio show_id = 0, show_name = null, next = 0, num_lines = 0,
show_locked = FALSE

get studio show_id
indi studio show_id = 1
get studio show_name
indi studio show_name = "Show 1"

get studio line_list
indi studio line_list =
[[IDLE, IDLE, null, "<sip:192.168.0.24>", null, TRUE, 1, null, ""],
[IDLE, IDLE, null, "<sip:21@192.168.0.24>", null, FALSE, 0, null, ""],
[IDLE, IDLE, null, "<sip:22@192.168.0.24>", null, FALSE, 0, null, ""],
[IDLE, IDLE, null, "<sip:23@192.168.0.24>", null, FALSE, 0, null, ""],
[IDLE, IDLE, null, "<sip:24@192.168.0.24>", null, FALSE, 0, null, ""],
[IDLE, IDLE, null, "<sip:40@192.168.0.23>", null, FALSE, 0, null, ""],
[IDLE, IDLE, null, "<sip:41@192.168.0.23>", null, FALSE, 0, null, ""],
[IDLE, IDLE, null, "<sip:42@192.168.0.23>", null, FALSE, 0, null, ""],
[NONE, IDLE, null, "<>", null, FALSE, 0, null, ""], [NONE, NONE, null, null,
null, FALSE, 0, 0, null], [NONE, NONE, null, null, null, FALSE, 0, 0, null],
[NONE, NONE, null, null, null, FALSE, 0, 0, null]]

# get number of lines in current studio
get studio num_lines
indi studio num_lines = 12
# get number of hybrids - both fixed and selectable
get studio num_hybrids
indi studio num_hybrids = 7
# get number of fixed hybrids (you can calculate number of selectable)
get studio num_hyb_fixed
indi studio num_hyb_fixed = 4

# hybrid list of names. Hybrid id is position in list
get studio hybrid_list
indi studio hybrid_list = ["Fixed Channel 1", "Fixed Channel 2", "Fixed
Channel 3", "Fixed Channel 4", "Selectable Channel 1", "Selectable Channel
2", "Selectable Channel 3"]

# get position of next line
get studio next
indi studio next = 0

# sets studio in busy state
busy_all studio state=TRUE

```

```
event studio busy_all = TRUE
event studio.line#1 state = BUSY, callstate = IDLE

get studio busy_all
indi studio busy_all = TRUE

# sends chat message to everybody in current studio
im studio message="DJ:Do you have more calls?"
event studio message = "DJ:Do you have more calls?"

get studio mute
indi studio mute = FALSE
get studio show_locked
indi studio show_locked = FALSE

#
# studio.line
#

get studio.line#1 state
indi studio.line#1 state = IDLE

get studio.line#1 callstate
indi studio.line#1 callstate = IDLE

get studio.line#1 name, local, remote
indi studio.line#1 name = null, local = "<sip:192.168.0.24>", remote = null

get studio.line#1 fixed
indi studio.line#1 fixed = TRUE
get studio.line#1 hybrid
indi studio.line#1 hybrid = 1

# RINGING_IN call
event studio next = 1
event studio.line#1 state = IDLE, callstate = RINGING_IN, time = 0

# get time since line RINGING_OUT, in milliseconds
get studio.line#1 time
indi studio.line#1 time = 50578
get studio.line#1 time
indi studio.line#1 time = 56108

# accepting RINGING_IN call and taking it on air
take studio.line#1
event studio next = 0
event studio.line#1 state = ON_AIR, callstate = ACCEPTED, time = 169040
event studio.line#1 state = ON_AIR, callstate = ESTABLISHED, time = 0

# comment for the line
set studio.line#1 comment="This is very interesting"
event studio.line#1 comment = "This is very interesting"
get studio.line#1 comment
indi studio.line#1 comment = "This is very interesting"

# seizes line
seize studio.line#2
event studio.line#2 state = SEIZED, callstate = IDLE, hybrid = 0, time =
null

# calling from line 2 to line 5 placing call on line 2 on-air, hybrid 5
call studio.line#2 number="sip:24@192.168.0.24" hybrid=2
event studio.line#2 state = SEIZED, callstate = RINGING_OUT, hybrid = 5,
time = 0
event studio next = 5
event studio.line#5 state = IDLE, callstate = RINGING_IN, hybrid = 0, time =
0
```

```

event studio.line#2 state = SEIZED, callstate = RINGING_OUT, hybrid = 5,
time = 290

# take next call on air
take studio.line
event studio.line#5 state = ON_AIR, callstate = ACCEPTED, hybrid = 5, time =
295969
event studio.next = 0
event studio.line#2 state = ON_AIR, callstate = ESTABLISHED, hybrid = 5,
time = 0
event studio.line#5 state = ON_AIR, callstate = ESTABLISHED, hybrid = 5,
time = 0

drop studio.line#5
event studio.line#5 state = IDLE, callstate = IDLE, hybrid = 0, time = null

# lock line so that it does not get dropped when conferencing
lock studio.line#1
event studio.line#1 state = ON_AIR_LOCKED, callstate = ESTABLISHED, time =
3610117

# unlocks line, puts back on air
take studio.line#1
event studio.line#1 state = ON_AIR, callstate = ESTABLISHED, time = 3702711

hold studio.line#1
event studio.next = 1
event studio.line#1 state = ON_HOLD, callstate = ESTABLISHED, time = 7992359

#mute ringers
# console 1 mutes
set studio.mute = TRUE
event studio.mute = TRUE
# console 2 mutes
set studio.mute = TRUE
event studio.mute = TRUE
# console 1 un-mutes
set studio.mute = FALSE
event studio.mute = TRUE
# console 2 un-mutes
set studio.mute = FALSE
event studio.mute = FALSE

# for VX phone to get date and time from VX Engine on startup
get cc.date
indi.cc.date = "2009-09-02T06:28:40"

# sets studio in auto answer mode - all calls will be answered and put on
hold
set studio.auto_answer = TRUE
event studio.auto_answer=TRUE
set studio.auto_answer = FALSE
event studio.auto_answer=FALSE

TODO: add some Handset examples

```

General

Port

TCP/IP port for accessing LWCP protocol on VX main unit is 20518

Event on changed

Whenever any of the properties of object is changed, event is sent only changed properties and their new values for that object. Some exceptions are, for instance line properties state and callstate are sent always together.

Comment

- comment. Any line that client sends starting with # is ignored by VX.

Case Sensitivity

Protocol is case sensitive

Protocol objects, their properties and methods

Property types:

string	text given in quotation marks, sample: "Show 1"
int	positive integer number, sample: 79
list	array of elements. Every element can be of different type, including list. In property descriptions list will be shown as values in square brackets. For each item will be given pseudo name and type, sample: [lines:int]
enum	enumerated constants, sample: IDLE ON_HOLD
bool	enum TRUE FALSE

Protocol objects:

cc - place where to put global methods and properties

studio - represents physical studio and its attributes

studio.line# - array of line objects for that studio. Line id is position in array

cc object

place where to put global methods and properties

get cc

studio_list list:[studio:[studio.id, studio.name]]

date string [YYYY-MM-DDThh:mm:ss]

returns date and time from VX Engine

login cc

user string

password string

you need to login before any commands can be executed. Default user = "user", password=""

status of success in ack:

ack cc logged=FALSE – login unsuccessful

ack cc logged=TRUE – login successful

set cc

mode enum: TALENT|PRODUCER

for compatibility with DD. Only write only property, no events. Functionality of other commands does not depend on this mode.

studio object

represents physical studio and its attributes. Current studio where client who is connected is located.

get studio

line array of line objects [line]

id int
current studio id

name string
current studio name
show_id int
id of show loaded in current studio. 0 if nothing loaded.
show_name string
num_lines int
number of lines in current studio
num_hybrids int
number of hybrids – both fixed and selectable
num_hyb_fixed int
number of fixed hybrids (you can calculate number of selectable)
next int
position of line that will be answered by next command (take studio.line)
busy_all bool
indicates if studio is in busy all state
mute bool
TRUE if ringers of all VX Directors set to muted
show_locked bool
TRUE indicates that show change for studio is forbidden
show_list list:[show:[show_id:int, show_name:string]]
list of shows available for this studio. Each list member is list consisting of:
show_id – id of show
show_name – name of the show
line_list list:[lines:[line.state, line.callstate, line.name?, line.local, line.remote,
line.fixed, line.hybrid, line.time?, line.comment]]
hybrid_list list:[hybrid_name:string]
list of hybrid names. Hybrid id is position in list.
hybrid_name – hybrid name as configured in VX

set studio

mute bool
mutes ringer of all VX Desktop Directors associated with this studio. Keeps track of how many times it was muted and need to un-mute the same number of times for ringers to start ringing. For example, if Console1 mutes ringer in Studio1, then Console2 mutes ringer in Studio1, then Console1 sends un-mute command – ringer still needs to be muted. When un-mute command is sent from second console, only then ringer is active again.
show_locked bool
if it is set to TRUE, show can not be changed for this studio. Future: maybe we want to implement "forced" parameter for select_show that can ignore locked state.
answer mode bool
sets studio in auto answer mode – all calls will be answered and put on hold

select studio

id int
selects current studio for this connection to work with. id – id of studio which to select as current. Triggers studio:event.

select_show studio

show_id int
loads show in this studio. All sessions will be affected. Triggers studio:event, line[:event]

busy_all studio

state bool[=TRUE]
sets studio in busy_all state or back to normal

im studio

message string
sends chat message to everybody in current studio. User name needs to be placed inside message with some separator, like ":". For example: "Prod: good work!"

Need to describe events!

line object

array of line objects for that studio. Line id is position in array. Each particular object represents one particular line in studio with its number and state.

get studio.line#x

state enum: IDLE|SEIZED|ON_AIR|ON_HANDSET|ON_HOLD|ON_HOLD_READY|ON_AIR_LOCKED|BUSY|SEIZED_EW|ON_HANDSET_EW|USED_EW|NONE
line state. Full state of line is formed together from state and call state. See diagram 2 and 3 for more details.

callstate enum: IDLE|DIALING|RINGING_OUT_OUT|RINGING_OUT_IN|ACCEPTING|ESTABLISHED
call state. Full state of line is formed together from state and call state. See diagram 2 and 3 for more details.

name string
name of line. Configurable in show configuration. By default client will display local phone number, but you can override that by some meaningful name. For example, "Hotline"

local string
local phone number. Number that line is assigned to and will start RINGING_OUT when somebody calls to this number.

remote string
remote or RINGING_IN number. Or caller ID. Number of caller that is calling in into this line.

fixed bool
fixed line or selectable line. Fixed lines has permanently assigned hybrids to them by show configuration. Selectable can specify hybrid every time it is taken on-air. This property will change when different show is loaded into this studio.

hybrid int
id of selected hybrid for this line. 0 means no hybrid selected

time int
time in milliseconds since last action with line. If `callstate = RINGING_IN`, since line started RINGING_OUT. If `state = ON_AIR`, since call is taken on air. If `state = IDLE`, since line is dropped.

comment string
comment for line. Can be set with set command.

set studio.line#x

comment string
comment to set.

seize studio.line#x

puts line in seize state. Before dialing, client would seize line thus reserving it for himself. Triggers state change for this line to `<state = SEIZED, callstate = IDLE>`

call studio.line#x

number string
[handset] bool=FALSE]
[hybrid] int=0]

calls to remote number given by number parameter. If handset is FALSE, places call on-air, otherwise on handset. If handset=FALSE, can specify also hybrid for selectable lines. For fixed lines no need to specify hybrid.

take studio.line[#x]

[handset] bool=FALSE]
[hybrid] int=0]

answers RINGING_IN call. If handset is FALSE, places call on-air, otherwise on handset. If handset=FALSE, can specify also hybrid for selectable lines. For fixed lines no need to specify hybrid.

If index of line is not given, then answers next call in line. If concrete line is specified with studio.line#x then answers that line. X is position of line from 1 to num_lines.

drop studio.line#x

drops line

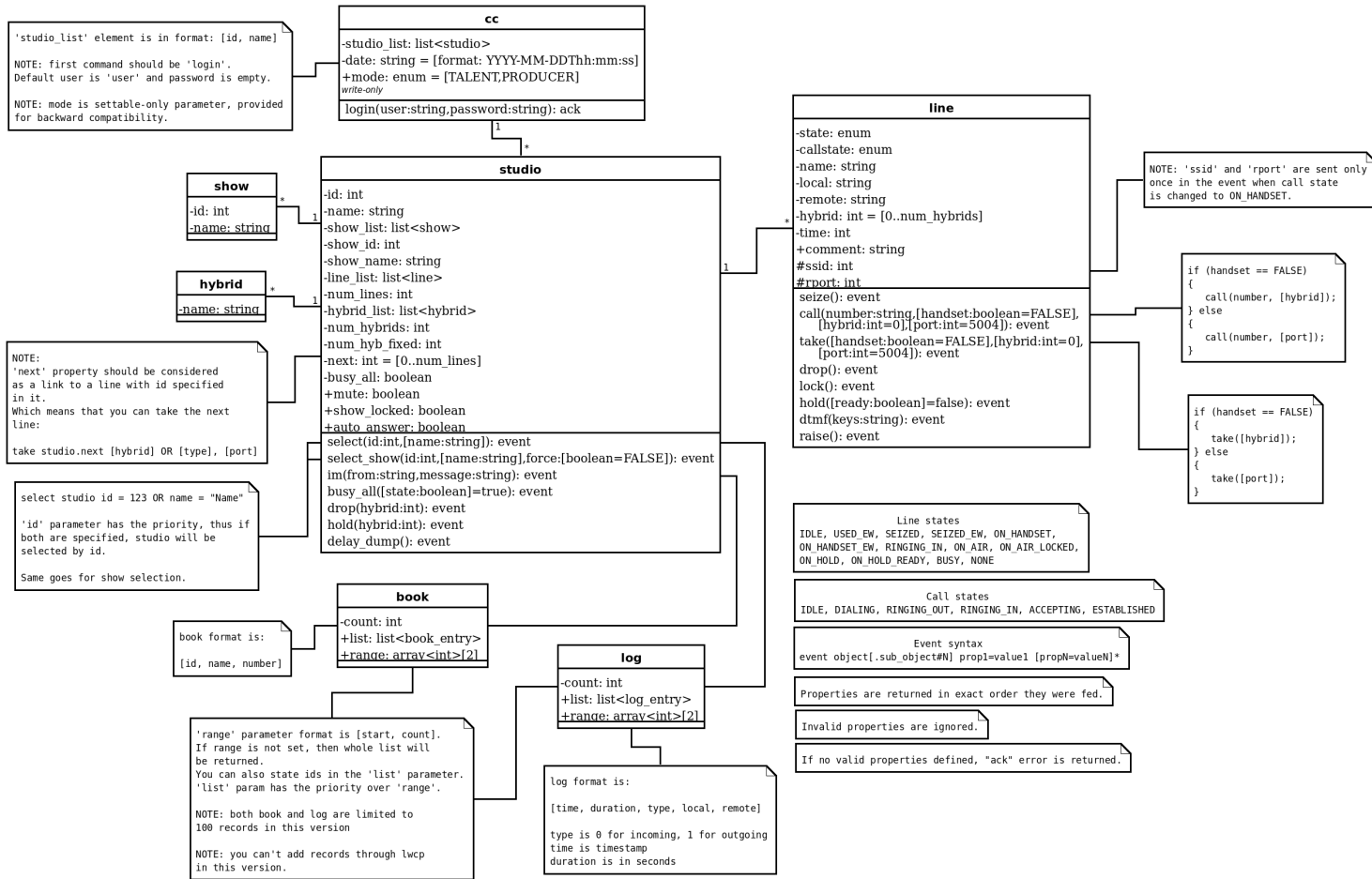
lock studio.line#x

lock line so that it does not get dropped when conferencing. Can be unlocked with take

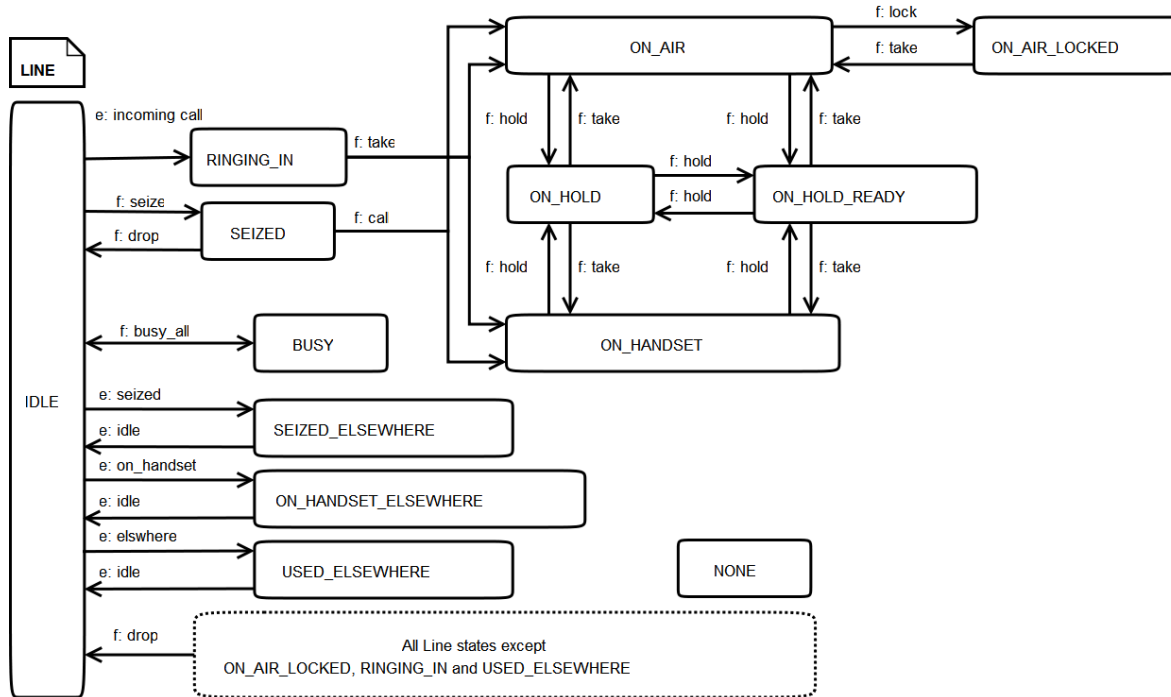
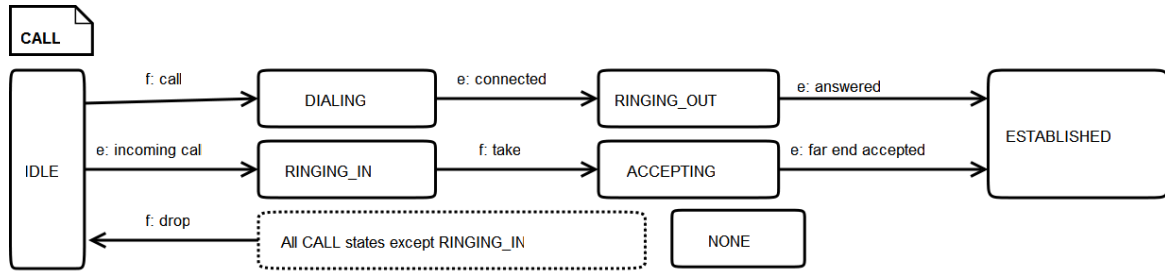
hold studio.line#x

[ready] bool=FALSE]

puts line on hold. If ready = TRUE, puts line in ready hold. Can be taken back on air or on handset with take.



1. LWCP objects, properties and functions



f - function called from LWCP
 e - event initiated from SIP or VX itself

2. Line and call state diagram

	IDLE	DIALING	RINGING_OUT	CALL RINGING_IN	ACCEPTING	ESTABLISHED
IDLE	IDLE	X	X	X	X	X
RINGING_IN	X	X	X	RINGING_IN	X	X
SEIZED	SEIZED	X	X	X	X	X
ON_AIR	X	DIALING	RINGING_OUT	X	PRE_ON_AIR	ON_AIR
ON_HANDSET	X	DIALING	RINGING_OUT	X	PRE_ON_HANDSET	ON_HANDSET
ON_HOLD	X	X	X	X	X	ON_HOLD
ON_HOLD_READY	X	X	X	X	X	ON_HOLD_READY
ON_AIR_LOCKED	X	X	X	X	X	LOCKED
BUSY	BUSY	X	X	X	X	X
SEIZED_EW	SEIZED_EW	DIALING_EW	RINGING_OUT_EW	X	X	X
ON_HANDSET_EW	X	HANDSET_EW	HANDSET_EW	X	HANDSET_EW	HANDSET_EW
USED_EW	USED_EW	X	X	X	X	X
NONE	NO_LINE	X	X	X	X	X

	State can be determined from LINE state only
	State can be determined from CALL state only

Icons

IDLE	dot	DIALING_EW	gray animated
RINGING_IN	ringing	RINGING_OUT_EW	dialpad
SEIZED	static dialpad	HANDSET_EW	gray handset
DIALING	animated dialpad	USED_EW	cross (X)
RINGING_OUT		NO_LINE	empty
PRE_ON_AIR	animated on air		
ON_AIR	on air		
PRE_ON_HANDSET	animated handset		
ON_HANDSET	handset		
ON_HOLD	on hold		
ON_HOLD_READY	on hold ready		
LOCKED	on air locked		
BUSY	busy		
SEIZED_EW	gray dialpad		