

LWCP sample commands that are implemented in Element

From AxiaWiki

```

# User module object MOD_USER
# Subobject BUT
# Full path to object MOD_USER#<module id>.BUT#<key number>
# "module id" is module rotary id number + 1
# KEY Properties
#   IND=<pattern index>|
#       OFF ind 0|
#       ON ind 255|
#       FLASH ind 4|
#       FLASH_ULTRA ind 1|
#       FLASH_FAST ind 2|
#       FLASH_MEDIUM ind 4|
#       FLASH_SLOW ind 8|
#       WINK ind 6|
#       FLASH_1000 ind 8|
#       FLASH_500 ind 4|
#       FLASH_250 ind 2|
#       FLASH_125 ind 1|
#       FLASH_375_125 ind 3|
#       DBL_WINK_125 ind 5|
#   KEY=UP|DOWN #read only
#   TEXT=<text literal> #can include /n to seperate lines. 3x6 char.
#   BACKCOLORON=[<red>,<green>,<blue>]|<#rrggbb str>|<color name str>
#   BACKCOLOROFF=[<red>,<green>,<blue>]|<#rrggbb str>|<color name str>
#
# BACKCOLORON and BACKCOLOROFF can have following format:
# W3C HTML 4.0 standard color in the form:
# #rrggbb or one of the 16 named colors:
# aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white,
#
# DISP Properties
#   IND=<the same values as for BUT.IND>
#   TEXT=<text literal> # if longer than 10, will scroll
#   AUTOSCROLL=NO|YES #default YES. If YES, text scrolls if longer than 10
#   ALIGN=LEFT|RIGHT|CENTER #default CENTER
#
# When some BUT is pressed, EVENT is generated, for example if BUT 1 is pressed, it sends:
#   EVENT MOD_USER#2.BUT#1 KEY=DOWN
#   EVENT MOD_USER#2.BUT#1 KEY=UP
# If GET comamnd is sent, INDI is sent back:
#   GET MOD_USER#2.BUT#5 KEY
# replays with
#   INDI MOD_USER#2.BUT#5 KEY=UP
#
# To execute test comamnds:
# telnet 127.0.0.1 4010
# All samples are for User module with protocol id 2 (h/w ID set to 1)
#
# button state
# if key 1 pressed
EVENT MOD_USER#2.BUT#1 KEY=DOWN
EVENT MOD_USER#2.BUT#1 KEY=UP
#
GET MOD_USER#2.BUT#1 KEY
INDI MOD_USER#2.BUT#1 KEY=UP
#
# button text
SET MOD_USER#2.BUT#1 TEXT="+-----+|Text|+-----+"
SET MOD_USER#2.BUT#2 TEXT=" Drop ", IND=ON
SET MOD_USER#2.BUT#3 TEXT="Line 1\nLine 2\nLine 3\n", IND=FLASH
SET MOD_USER#2.BUT#5 TEXT="Press\nMe", LED=WINK
SET MOD_USER#2.BUT#6 TEXT="Active", IND=ON
#
#flash, ON, OFF
SET MOD_USER#2.BUT#1 IND=FLASH_ULTRA
SET MOD_USER#2.BUT#2 IND=FLASH_FAST

```

```
SET MOD_USER#2.BUT#3 IND=FLASH_MEDIMU
SET MOD_USER#2.BUT#4 IND=FLASH_SLOW
SET MOD_USER#2.BUT#5 IND=WINK
SET MOD_USER#2.BUT#6 IND=ON
SET MOD_USER#2.BUT#7 IND=OFF
SET MOD_USER#2.BUT#8 IND=FLASH
SET MOD_USER#2.BUT#9 TEXT="OK", IND=DBL_WINK_125

# colors
SET MOD_USER#2.BUT#1 BACKCOLORON=[255,0,0], BACKCOLOROFF=[0,255,0]
SET MOD_USER#2.BUT#2 BACKCOLORON="purple", BACKCOLOROFF="yellow"
SET MOD_USER#2.BUT#3 TEXT="Hi", BACKCOLORON="#33bbff", BACKCOLOROFF="#ff3300", IND=FLASH_SLOW
SET MOD_USER#2.BUT#4 TEXT="Hi", BACKCOLORON="red", BACKCOLOROFF="green", IND=DBL_WINK_125

# character display (10 char)
SET MOD_USER#2.DISP TEXT="Hello \nWorld", IND=FLASH_SLOW #will scrooll as longer than 10
SET MOD_USER#2.DISP TEXT="Hello World", AUTOSCROLL=NO, IND=FLASH_SLOW #will NOT scrooll
SET MOD_USER#2.DISP TEXT="Show", IND=ON
SET MOD_USER#2.DISP TEXT="Show", IND=ON, ALIGN=RIGHT
SET MOD_USER#2.DISP TEXT="Show", IND=ON, ALIGN=LEFT
SET MOD_USER#2.DISP TEXT="Show", IND=ON, ALIGN=CENTER

#VMIX
SET VMIX.SUB#1.IN#1 Gain=80
EVENT VMIX.SUB#1.IN#1 Gain=80

SET VMIX.SUB#1.IN#5 Gain=90
EVENT VMIX.SUB#1.IN#5 Gain=90

GET VMIX.SUB#1.IN#5 State, Gain, TimeUp, TimeDown
INDI VMIX.SUB#1.IN#5 State=OFF, Gain=-200, TimeUp=50, TimeDown=50

SET VMIX.SUB#1.IN#5 State=ON
EVENT VMIX.SUB#1.IN#5 State=ON

#show profile change
# change event indication
EVENT AppControl ShowProfID=-16318101, ShowProfName="Bob Show", ShowProfStat=LOADING
EVENT AppControl ShowProfID=-16318101, ShowProfName="Bob Show", ShowProfStat=READY
# get current profile
GET AppControl ShowProfID,ShowProfName,ShowProfStat
INDI AppControl ShowProfID=-16318094, ShowProfName="Morning Show", ShowProfStat=READY
# get profile list
GET AppControl ShowProfList
INDI AppControl ShowProfList=%BeginEncap%
<List>
  <ShowProfile><ID>-16318101</ID><Name>Bob Show</Name></ShowProfile>
  <ShowProfile><ID>-16318261</ID><Name>Show1</Name></ShowProfile>
  <ShowProfile><ID>-16318149</ID><Name>Show2</Name></ShowProfile>
  <ShowProfile><ID>-16318094</ID><Name>Morning Show</Name></ShowProfile>
</List>
%EndEncap%

# fader control and indication
EVENT LwCH#501 OFF_But=Down,ON_State=On
EVENT LwCH#501 ON_State=OFF
EVENT LwCH#501 OFF_But=Up,ON_State=Off

# turn fader ON
SET LwCH#501 ON_State=ON
EVENT LwCH#501 ON_State=ON

# turn fader OFF
SET LwCH#501 ON_State=OFF
EVENT LwCH#501 ON_State=OFF

# alternative: addressing by fader number:
# ON
SET FaCH#1 ON_State=ON
EVENT FaCH#1 ON_State=ON

# OFF
SET FaCH#1 ON_State=OFF
EVENT FaCH#1 ON_State=OFF

# Mute Mic and restore, indication - sent also if mic muted from console or GPIO
SET LwCH#501 Mute_State=MUTED
EVENT LwCH#501 Mute_State=MUTED
SET LwCH#501 Mute_State=NORMAL
EVENT LwCH#501 Mute_State=NORMAL
```

```
# Motorized fader control. Works also for non-motorized faders. Implemented since 2.0.x
# SET LwCH#<lpid> Fader_Gain=<value in dB>
SET LwCH#501 Fader_Gain=-30

# get current fader gain
GET LwCH#1439 Fader_Gain
INDI LwCH#1439 Fader_Gain=-6.7

# get initial state
GET LwCH#1439 Fader_State
INDI LwCH#1439 Fader_State=DOWN

# event if if/when fader is over "Fader start" threshold
EVENT LwCH#1439 Fader_State=UP
EVENT LwCH#1439 Fader_State=DOWN

# PGM1 - PGM4 (would also work with LwCH)
SET FaCH#1 Asg_PGM2=ON, Asg_PGM3=ON, Asg_PGM4=OFF
EVENT FaCH#1 Asg_PGM2=ON
EVENT FaCH#1 Asg_PGM3=ON
EVENT FaCH#1 Asg_PGM4=OFF

# PREVIEW (would also work with LwCH)
SET FaCH#1 Asg_PREV=ON
EVENT FaCH#1 Asg_PREV=ON

# Show profile control
# Get profile list
# -----
GET AppControl ShowProfList
INDI AppControl ShowProfList=%BeginEncap%
<List>
  <ShowProfile><ID>28</ID><Name>777</Name></ShowProfile>
  <ShowProfile><ID>-3</ID><Name>Default show profile</Name></ShowProfile>
  <ShowProfile><ID>27</ID><Name>mon_vol</Name></ShowProfile>
  <ShowProfile><ID>25</ID><Name>new1</Name></ShowProfile>
  <ShowProfile><ID>26</ID><Name>new2</Name></ShowProfile>
  <ShowProfile><ID>29</ID><Name>Talkback def</Name></ShowProfile>
</List>
%EndEncap%

# Get current profile identity and status
# -----
GET AppControl ShowProfID,ShowProfName,ShowProfStat
INDI AppControl ShowProfID=25, ShowProfName="new1", ShowProfStat=READY

# Get current profile and profile list in one command
# -----
GET AppControl ShowProfID,ShowProfName,ShowProfStat,ShowProfList
INDI AppControl ShowProfID=25, ShowProfName=new1, ShowProfStat=READY, ShowProfList=%BeginEncap%
<List>
  <ShowProfile><ID>28</ID><Name>777</Name></ShowProfile>
  <ShowProfile><ID>-3</ID><Name>Default show profile</Name></ShowProfile>
  <ShowProfile><ID>27</ID><Name>mon_vol</Name></ShowProfile>
  <ShowProfile><ID>25</ID><Name>new1</Name></ShowProfile>
  <ShowProfile><ID>26</ID><Name>new2</Name></ShowProfile>
  <ShowProfile><ID>29</ID><Name>Talkback def</Name></ShowProfile>
</List>
%EndEncap%

# Observe profile changes
# -----
EVENT AppControl ShowProfID=25, ShowProfName="new1", ShowProfStat=LOADING
EVENT AppControl ShowProfID=25, ShowProfName="new1", ShowProfStat=READY

# Setting profile
# -----
SET AppControl ShowProfID=28
EVENT AppControl ShowProfID=28, ShowProfName="777", ShowProfStat=LOADING
EVENT AppControl ShowProfID=28, ShowProfName="777", ShowProfStat=READY

# LWCP commands over Multicast (for Accessory modules)
# -----
# Multicast address used: 239.192.255.4
#
# object is LwCH#<lw channel number>
#
# module -> Element (commands sent and received at port 4011):
```

```

# BTN_ON = {DOWN|UP}
# BTN_OFF = {DOWN|UP}
# BTN_TALK = {DOWN|UP}
# BTN_MUTE = {DOWN|UP}
# BTN_HPpset1 = {DOWN|UP}
# BTN_HPpset2 = {DOWN|UP}
# BTN_HPsel = {DOWN|UP}
# ROT_HPsel = delta
#
# Element -> Module (commands sent and received at port 4012)
# LMP_ON = {ON|OFF}
# LMP_OFF = {ON|OFF}
# LMP_TALK = {ON|OFF}
# LMP_MUTE = {ON|OFF}
# LMP_HPset1 = {ON|OFF}
# LMP_HPset1 = {ON|OFF}
# DSP_HPtext = "text"

=====
Added in version 2.2 - RQ 3692
=====

// Getting source profile list for an individual fader:
C: get FaCH#3 src_list
S: indi FaCH#3 src_list=%beginencap%
<list>
  <src>
    <id>1</id>
    <name>Goog_115</name>
    <lwch>115</lwch>
  </src>
  <src>
    <id>2</id>
    <name>Goog_116</name>
    <lwch>116</lwch>
  </src>
</list>
%endencap%

// Checking the actually loaded source profile. A source is loaded on fader 3, in full control mode:
C: get FaCH#3 src_id, src_name, src_lwch, src_stat
S: indi FaCH#3 src_id=1, src_name="Goog_115", src_lwch=115, src_stat=22

// Changing the source. Source loading/unloading is followed by an event notification, regardless of what
C: set FaCH#3 src_id=2
R: Load the requested source
S: event FaCH#3 src_id=2, src_name="Goog_116", src_lwch=116, src_stat=21

// Unloading the source:
C: set FaCH#3 src_id=-1
R: Unload the source
S: event FaCH#3 src_id=-1

// Source change notifications. The operations were initiated not in terms of this LWCP session - by the
Operator/other LWCP: Request a source change on fader#3
R: Load the requested source on fader#3
S: event FaCH#3 src_id=2, src_name="Goog_116", src_lwch=116, src_stat=21

Operator/other LWCP: Request unloading the source from fader#4
R: Unload source from fader#4
S: event FaCH#4 src_id=-1

=====
Preliminary Intercom module control
=====
# list of objects and properties used
# object Intercom
# property 'type' - value: {-1,type} type of intercom module installed, ReadOnly
# value derived straight from module sub-type: 1 - 10 OLED module, 2 - 5 OLED module, 3 - Keys only,
# subobject Intercom.key - keys on intercom type 0 module (without OLED) with 2 color LEDs
# property 'KEY' - value: {UP,DOWN} state of a key, ReadOnly
# property 'BACKCOLOROFF' - value: {OFF,BLACK,RED,GREEN,YELLOW} color of key led, WriteOnly
# property 'BACKCOLORON' - value: {OFF,BLACK,RED,GREEN,YELLOW} color of key led, WriteOnly
# property 'IND' - value: {OFF,ON,...other ind_styles...} style of key led, WriteOnly
# subobject Intercom.btn - small buttons with leds next to them, also Rotary knobs
# property 'KEY' - value: {UP,DOWN} state of a button, ReadOnly
# property 'rot' - value: {+/-delta} relative rotation, EventOnly
# property 'IND' - value: {OFF,ON,...other ind_styles...} style of key led, WriteOnly
# subobject Intercom.oled - oleds

```

```
# property 'line1'          - value: {any} text presented on top line of OLED, WriteOnly
# property 'line2'          - value: {any} text presented on bottom line of OLED, WriteOnly

# Work example:
# Connect to Element:4010
# Get Type of Intercom module installed
C: GET Intercom type
S: INDI Intercom.type=2
# Subscribe to Intercom events and enable SET commands from Client (GET commands will work even without t
C: SUB Intercom
S: ACK Intercom OP=SUB,Status=OK

# begin to receive Intercom Events if something will happen
# Filmcap buttons:
S: EVENT Intercom.key#1 KEY=DOWN
S: EVENT Intercom.key#1 KEY=UP
# OLED module small talk and listen buttons:
S: EVENT Intercom.btn#1 KEY=DOWN
S: EVENT Intercom.btn#1 KEY=UP
# Rotary knobs
S: EVENT Intercom.btn#129 rot=1
S: EVENT Intercom.btn#129 rot=-1
S: EVENT Intercom.btn#129 KEY=DOWN
S: EVENT Intercom.btn#129 KEY=UP

# set indications
# OLED module small talk and listen buttons:
C: SET Intercom.btn#1 IND=ON
# Filmcap buttons:
C: SET Intercom.key#1 BACKCOLOROFF=RED,BACKCOLORON=GREEN,IND=FLASH_FAST
# OLED displays:
C: SET Intercom.oled#1 line1="text",line2="example"
C: SET Intercom.menu list=["IC LIME","IC YELLOW","Music IFB2","News IFB1"],selected=4

# Unsubscribe (not needed if client is intended to close connection)
C: UNSUB Intercom
S: ACK Intercom OP=UNSUB,Status=OK
```

Retrieved from "http://www.axia.lv/wiki/index.php/LWCP_sample_commands_that_are_implemented_in_Element"

- This page was last modified on 25 May 2011, at 14:05.