|  |
| --- |
| **LCL Guida ufficiale BETA 1.1.0 rev 01-01-2010** |
| Light Command Line beta 1.1.0 rev 1-1-2010 |
|  |
| Guida per l'utilizzo di LCL, software per registrare o modificare i file creati da LightWall BETA 16.0.1 E' noto un bug sulla gestione del percorso che impedisce la registrazione partendo da file completamente vuoti. |
|  |
| **Ks89** |
| **01/01/2010** |
|  |

# 1. Descrizione struttura, comando e valore

## 1.1 Struttura

## 1.2 Comando

## 1.3 Valore

## 2. Tutte le strutture e i comandi possibili 2.1 Struttura "write"

## 2.2 Struttura "lic"

## 2.3 Struttura "writepc"

## 2.4 Struttura "pathw"

## 2.5 Struttura "license"

## 2.6 Struttura "save"

## 2.7 Struttura "restore"

## 2.8 Struttura "view"

## 2.9 Comando speciale: "matrice"

# 3. Eseguire una registrazione manuale (working progress!) A causa di un bug, non si può registrare da zero, cioè dai file vuoti, perchè il percorso viene gestito in modo errato. Forse per colpa di LCL o forse per colpa di LightWall (molto probabile LightWall) Descrizione bug: il percorso salvato sui file e quello generato ogni volta da LightWall per qualche strano motivo non coincidono. Forse è dovuto al fatto che il percorso generato non presenta un '\0' alla fine, che indica la fine della stringa.

# 1 Descrizione struttura, comando e valore

## 1.1 Struttura

Struttura è il campo dalla quale si diramano i comandi. Ognuna di esse possiede solo una serie limitata di comandi possibili.

## 1.2 Comando

Comando è il sottolivello di struttura. Ne esistono una serie limitata con particolari funzionalità ed applicazioni, indicate successivamente

## 1.3 Valore

Ultimo campo. In base alla struttura e al valore diventa il valore stesso nel file. Cioè il valore inserito viene copiato all'interno del campo della "struct del file" per poi essere salvato con l'apposita struttura "save".

# 2 Tutte le strutture e i comandi possibili 2.1 Struttura "write"

**write** : struttura associata al file ks.dat  
**struttura.comando. valore**

write.codice ( 7 numeri al massimo 0->9)

write.controllo (inserire codice, controllo viene generato in automatico)

write.reg (jhfiusq (registrato) o jhfiusw(non registrato))

write.inutile (5 numeri : 12344 (file non corrotto))

write.inutile2 (inserire 'y' o 'n'. Se y=-546,451405 cioè file NON corrotto)

write.pcname (al massimo di 20 caratteri)

write.space (inutile, massimo 2 caratteri)

write.path (al massimo 100 caratteri )

write.controllo\_registrazione (se 22: non registrato, se 32: registrato)

write.controlloic (7 numeri al massimo 0->9)

write.mixer (sempre e solo 1485170210, il prog separa i numeri in 1 4 8 5 1 7 0 2 10)

2.2 Struttura "lic"

**lic** : struttura associata al file lic.dat  
**struttura.comando. valore**

lic.codice\_dat ( 7 numeri al massimo 0->9. DEVE ESSERE UGUALE A write.codice)

lic.controllo\_dat (inserire codice\_dat. DEVE ESSERE UGUALE A write.codice)

lic.reg (1234667 (registrato) o 1234567(non registrato))

lic.inutile (5 numeri : 12344 (file non corrotto))

lic.inutile2 (inserire 'y' o 'n'. Se y=-546,451405 cioè file NON corrotto)

lic.pcname (al massimo di 20 caratteri)

lic.controllo\_registrazione (se 22: non registrato, se 32: registrato)

lic.controlloic (7 numeri al massimo 0->9. Generato a caso in LightWall)

lic.codice\_a (7 numeri al massimo 0->9. Nessuna condizione\*)

lic.controllo\_a (inserire codice\_a. Nessuna condizione)

*\*Il programma genera in automatico un codice valido cercando di tenere il più possibile i valori inseriti dall'utente. Spesso però li deve sostituire per far si che possa funzionare.*  
  
2.3 Struttura "writepc"  
**writepc** : struttura associata al file rek.dll  
**struttura.comando. valore**writepc.pcname (al massimo di 20 caratteri)  
writepc.controlloic (7 numeri al massimo 0->9)  
writepc.hash (qualunque valore. md5 e sha-1 vengono generati in automatico e non in base al valore inserito)  
  
2.4 Struttura "pathw"  
**pathw** : struttura associata al file irr.dat  
**struttura.comando. valore**pathw.path (al massimo 100 caratteri )  
pathw.controlloic (7 numeri al massimo 0->9)  
pathw.cont\_licenza (valore da 0->3. Indica il numero di tentativi fatti senza successo per attivare la licenza. Questo dato viene incrementato fino a 3 dopo 3 errori alla domanda della licenza. A 3 LightWall richiede lo sblocco licenza)

2.5 Struttura "license"

**license** : struttura associata al file license.lic   
**struttura.comando. valore**  
license.nome (Inserire nome a cui è associata la licenza [non ammessi spazi])

license.cogn (Inserire cognome a cui è associata la licenza [non ammessi spazi]))

license.soc (Inserire società a cui è associata la licenza [non ammessi spazi]))

license.mail (Inserire e-mail a cui è associata la licenza [non ammessi spazi]\*)

license.reg (jhfiusq (registrato) o jhfiusw(non registrato)\*\*)

license.inutile (5 numeri : 12344 (file non corrotto))

license.inutile2 (inserire 'y' o 'n'. Se y=-546,451405 cioè file NON corrotto)

license.num\_domanda (numero da 1 a 6)

license.domanda (risposta alla domanda [50 caratteri al massimo. Spazi ammessi])

*\*Deve essere valida. Composta così: x@a.x  
\*\*Il valore deve essere uguale a quello di write.reg*

2.6 Struttura "save"

**save** : struttura che permette di salvare i file  
**struttura.comando. valore**  
save.ks (Qualunque. Il comando esegue il salvataggio dei dati precaricati)

save.rek (Qualunque. Il comando esegue il salvataggio dei dati precaricati)

save.irr (Qualunque. Il comando esegue il salvataggio dei dati precaricati)  
save.lic (Qualunque. Il comando esegue il salvataggio dei dati precaricati)  
save.license (Qualunque. Il comando esegue il salvataggio dei dati precaricati)

2.7 Struttura "restore"

**restore** : struttura che ripristina ai valori di fabbrica i file (sostituisce Ripristino fabbrica beta)  
**struttura.comando. valore**  
save.ks (Qualunque. Il comando esegue il ripristino fabbrica del file indicato)

save.rek (Qualunque. Il comando esegue il ripristino fabbrica del file indicato)

save.lic (Qualunque. Il comando esegue il ripristino fabbrica del file indicato)

2.6 Struttura "view"  
**view** : struttura per visualizzare le informazioni contenute nella memoria del programma e nei file  
**struttura.comando. valore**view.ks (Mostra i dati nella struttura del programma e quelli sul file)

view.rek (Mostra i dati nella struttura del programma e quelli sul file)

view.irr (Mostra i dati nella struttura del programma e quelli sul file)   
view.lic (Mostra i dati nella struttura del programma e quelli sul file)  
view.license (Mostra i dati nella struttura del programma e quelli sul file)

view.all (Carica nel programma tutti i dati sui file)

N.B.  
all, riempie tutti campi delle struct del programma con i dati letti dei file, sovrascrivendo tutti i valori non salvati precedentemente. E' un comando irreversibile.  
Gli altri comandi mostrano il contenuto dei campo delle struct del software e subito sotto il file.  
Notare, che una volta eseguito questo comando, tutti i dati nei campi del programma vengono sovrascritti con quelli sui file. Quindi prima di eseguirlo salvare il file con i dati inseriti in precedenza.

2.6 Comando speciale: "matrice"

**matrice:** è un particolare comando, che non ha ne struttura ne valore, per generare la matrice su tutti i file.

xxxx.matrice.xxxx -> qualunque struttura e qualunque valore non influenzano la matrice.  
Essa viene generata in automatico dal programma dopo aver eseguito il comando view.all.1.  
Infatti, è importante che tutti i campi necessari siano stati pre-caricati. Essendo moltissimi, conviene usare il view.all.1

# 3. Eseguire una registrazione manuale

ATTENZIONE: questo sistema esegue la registrazione mantenendo nomepc, path e vari codici.  
E' possibile cambiare tutto. Basta fare attenzione a rispettare le istruzioni date sopra.

**IL PERCORSO IN QUESTA VERSIONE NON PUO' ESSERE CAMBIATO A CAUSA DI UN BUG ANCORA NON RISOLTO DEFINITIVAMENTE**

1) Posizionare i file "LCL beta 1.1.0.exe" e "libeay32.dll" nella cartella di installazione di LightWall  
2) Eseguire "LightWall.exe". Chiuderlo subito dopo.  
3) A questo punto si saranno generati i file aggiuntivi necessari.  
4) Eseguire LCL beta 1.1.0 e digitare i seguenti comandi in ordine

LISTA COMANDI:

>> view.all.1 //carica tutti i dati dai file al programma

>> view.ks.1 //mostra il contenuto di ks.dat

>> write.reg.jhfiusq //imposta reg=jhfiusq, cioè REGISTRATO

>> write.controllo\_registrazione.32 //imposta 32, cioè REGISTRATO

>> view.lic.1 //mostra il contenuto di lic.dat

>> lic.reg.1234667 //imposta 12345667, cioè REGISTRATO

>> lic.controllo\_registrazione.32 //imposta 32, cioè REGISTRATO

>> view.rek.1 //mostra il contenuto di rek.dll

>> writepc.hash.1 //genera md5 e sha-1 (non è necessario, ma per sicurezza)

>> save.ks.1 //salva ks.dat

>> save.rek.1 //salva rek.dll

>> save.lic.1 //salva lic.dat

>> a\_caso.matrice.1 //genera la matrice per tutti i file. vedi 2.6

>> save.ks.1 //salva ks.dat

>> save.rek.1 //salva rek.dll

>> save.lic.1 //salva lic.dat

>> exit //esce dal programma ('exit' o 'quit' causano l'uscita)