

HSYCO SERVER

MANUALE DI CONFIGURAZIONE E PROGRAMMAZIONE

HSYCO 3.1.2

DOC REL. IT 1

© 2007-2012 Home Systems Consulting SpA. Tutti i diritti riservati.

HSYCO® Manuale di Configurazione e Programmazione

La riproduzione, copia con qualsiasi mezzo elettronico o meccanico o traduzione non autorizzate sono proibite.

I contenuti di questa guida sono protetti dalle leggi e dai trattati internazionali sul copyright, anche se essa non è distribuita con un software accompagnato da una licenza per l'utente finale. Questa guida è fornita unicamente per scopi informativi e può essere soggetta a modifiche senza preventiva comunicazione.

Home Systems Consulting nega espressamente ogni responsabilità su eventuali errori contenuti in questa guida.

Installazione	19
Accesso al Sistema Operativo	20
La Console di Configurazione	21
Pagine di Amministrazione	23
Gestione degli Account	26
Password di Sistema	27
Impostazioni di Rete	29
Impostazioni dell'orologio	31
Il Manager	32
Il File Manager	34
Aggiornamento e riavvio di HSYCO	36
Il Log Viewer	37
Organizzazione dei File	38
hsyco/access.ini	39
hsyco/acl.ini	39
hsyco/console.log	39
hsyco/hsyco.ini	40
hsyco/hsyco.jar	40
hsyco/hsyco.keys	40
hsyco/keys.data	40
hsyco/license.txt	41
hsyco/systemtopo.txt	41
hsyco/timers.data	41
hsyco/com/hsyco	42

hsyco/www	42
hsyco/www/<project>/index.hsm	42
hsyco/www/<project>/img	43
hsyco/www/<skin>	43
hsyco/www/img	43
hsyco/www/public	43
hsyco/motion	44
hsyco/logs	44
hsyco/ir	45
hsyco/dataloggers.ini	45
hsyco/data	47
hsyco/Scripts	47
La Chiave di Licenza Software	48
Accesso al Sistema Operativo	50
Avviare e Fermare HSYCO	51
Accesso ai File	52
Accesso ai File da Microsoft Windows	52
Accesso ai File da Mac OS X	55
Configurazione	57
Sicurezza e Gestione Utenti	58
Impostazioni Iniziali di hsyco.ini	59
Introduzione ai Sistemi di Campo	61
Il File hsyco.ini	64
Controllo Accessi	64
Livelli di Log	67

Alta Affidabilità	68
Rete	69
Orologio	70
DMX Gateway	72
Telecamere	73
I/O Server	79
Server HSYCO Remoti	80
Porte di Comunicazione Seriale	81
IRTrans	83
Squeezebox	84
Public Announcement	85
Timer	85
Servizi di Localizzazione	86
Data Loggers	87
Email	90
Altri Parametri	91
Personalizzazione	93
Il Database di Configurazione	94
(location)	95
(devices)	96
Le Pagine Web	98
Il Project Editor	99
Creare un Nuovo Progetto	100
Layout e Coordinate delle Pagine	100
Modificare un Progetto	102

Menu, Pagine e Pop-Up	104
Gli Oggetti di una Pagina	105
Container	107
Il file sorgente del progetto index.hsm	108
Objects Reference	109
Oggetti	109
Direttive Generali	110
(#skin <name>)	110
(#language <language id>)	110
(#size <page size>)	111
(#scale <factor>)	111
(#kiosk <mode>)	112
(#locked <mode>)	112
(#deviceimage disable)	112
(#location <list>)	112
(#cameralist <list>)	113
(#cameragridlist <list>)	113
(#cameraoverlay <overlay list>)	114
(#include <file name>)	114
Menù Principale	114
(header <title>)	114
(menu)	115
(menu#landscape)	115
(menu#portrait)	115
(endofmenu)	115
Pagine	116
(link <pos>; <color>; <page>; <label>)	116

(linkmini <pos>; <color>; <page>; <label>)	116
(linkmicro <pos>; <color>; <page>; <label>)	116
(dlink <pos>; <color>; <page>; <label>)	117
(page <id>; <description>; <protection>)	118
(page#landscape <id>; <description>; <protection>)	118
(page#portrait <id>; <description>; <protection>)	118
(popup <id>; <description>)	119
(endofpage)	119
(container <pos>; <visibility>)	119
(endofcontainer)	120
(selector <file>; <pos>; <width>; <height>; <container id>; <group>; <text>; <css>)	120
(background <file>)	121
Testi	121
(text <pos>; <text>; <css>)	121
(marquee <pos>; <width>; <height>; <dir>; <speed>; <text>; <css>)	122
Elementi Grafici	122
(panel <pos>; <width>; <height>; <color>)	122
(hbar <pos>; <width>)	123
(vbar <pos>; <height>)	123
(image <file>; <pos>; <width>; <height>; <text>; <css>)	123
(imagelink <file>; <pos>; <width>; <height>; <page>; <text>; <css>)	124
(chartlid <pos>; <width>; <height>; <attributes>)	125
(datalogger <id>; <pos>; <width>; <height>; <label>; <attributes>)	127
(video <src>; <pos>; <width>; <height>; <mode>)	129
Amministrazione	130
(adminlink <function>; <pos>; <color>; <label>)	130
Luci e Automazioni	131

(button <address>; <pos>; <label>)	131
(3button <address>; <pos>; <label>)	131
(buttonicon <address>; <pos>; <color>)	132
(buttonimage <address>; <pos>; <width>; <height>; <unknown img>; <on img>; <off img>; <up img>; <down img>; <offup img>; <offdown img>; <text>; <css>)	133
(dimmer <address>; <pos>; <label>)	134
(dmx <address>; <pos>; <label>)	134
(dmxrgb <address>; <pos>; <label>)	135
Controllo Temperatura	136
(temp <server>; <address>; <label>; <pos>)	136
(tempmini <server>; <address>; <label>; <pos>)	137
Timer e Scheduler	137
(timer <id>; <pos>; <label>)	137
(scheduler <pos>; <width>; <height>; <mode>; <names>)	138
Telecamere	140
(cameralink <cam id>; <pos>; <color>; <label>; <dest panel id>)	140
(camera <cam id>; <pos>; <width>; <height>; <dest panel id>)	140
(camerapanel <cam id>; <pos>; <width>; <height>; <cam list>)	141
IRTrans	142
(ir <irid>; <com>; <pos>; <color>; <label>)	142
(irmini <irid>; <com>; <pos>; <color>; <label>)	142
(irmicro <irid>; <com>; <pos>; <color>; <label>)	142
Player musicali	143
(music <id>; <pos>; <label>)	143
(musicsync <pos>)	143
(nuvo <id>; <zone>; <pos>)	144
(nuvomini <id>; <zone>; <pos>)	144
Tasti USER	145

(user <name>; <param>; <pos>; <color>; <label>)	145
(usermini <name>; <param>; <pos>; <color>; <label>)	145
(usermicro <name>; <param>; <pos>; <color>; <label>)	145
(userrgb <name>; <param>; <pos>; <color>; <label>)	146
(userimage <file>;<pos>;<width>;<height>;<name>;<param>; <text>; <css>)	147
(slider!id <name>; <pos>; <label>)	148
(sliderbutton!id <name>; <pos>; <label>)	149
Form	150
(input!id <pos>; <css>)	150
(submit!id <pos>; <color>; <label>)	150
(submitmini!id <pos>; <color>; <label>)	150
(submitmicro!id <pos>; <color>; <label>)	150
(submitimage <file>; <pos>; <width>; <height>; <text>; <css>)	151
(keypad!id <pos>; <min>; <max>; <digits>; <decimals>; <type>; <label>; <css>)	152
Weather	153
(weather!id <pos>)	153
Le Opzioni di Visualizzazione	154
Lingua delle Pagine Web	155
Il Funzionamento dell'Interprete di Skin	156
Inserire Codice HTML nelle Pagine	158
Programmazione a Eventi	159
EVENTS	160
Il linguaggio EVENTS	161
Commenti	162
Altre Regole Generali	162
Eventi Stabili e Transitori	163

Un Evento, più Regole	165
Azioni	166
Variabili	166
Variabili e Azioni	167
Variabili e Eventi	169
Variabili Volatili e Persistenti	170
Variabili Predefinite	170
Eventi	172
I/O Server	172
IO	172
IOSTART	173
DMX	174
DMXSTART	174
DMX	175
Telecamere	176
CAMERA	176
CAMERACOMMAND	177
Antintrusione	178
SECURITY	178
IRTrans	179
IR	179
Squeezebox	180
MUSIC	180
PBX	181
PBX	181
Servizi di Rete	182

LOCATION	182
PING	182
URL	183
Timer e Scheduler	184
TIMER	184
Leak Detector	185
LEAK	185
Eventi Interni	186
HSYCOSTART	186
USER	186
PROGRAMTIMER	187
TIME	187
DAY	188
NIGHT	188
SUNAZIMUTH	189
SUNELEVATION	189
POWER	190
Azioni	191
I/O Server	191
IO	191
DMX	192
Telecamere	193
CAMERA	193
CAMERAREC	193
CAMERARECFULL	194
Porte di Comunicazione Seriale	195
COMM	195

IRTrans	196
IR	196
Squeezebox	197
MUSIC	197
Public Announcement	198
AUDIO	198
Mail	200
MAIL	200
Log	202
LOG	202
Servizi di Rete	203
PING	203
URL	203
Data Logger	205
DATALOGGER	205
Leak Detector	207
LEAK	207
WAIT	209
USER	209
POWER	210
PROGRAMTIMER	210
UISET	212
Programmazione Java	219
Costanti Predefinite	220
Metodi Callback	221
I/O Server	221

public static void IOStartupEvent(int serverIndex)	221
public static void IOEvent(String id, String value)	221
Timer e Scheduler	223
public static boolean UserTimerEvent(String name, boolean active)	223
DMX	224
public static void DmxStartupEvent(int serverIndex)	224
public static void DmxEvent(int channel, int state)	224
public static int DmxFilter(int channel, int state, boolean reverse)	224
Telecamere	226
public static void CameraMotionEvent(String eventName, long remoteTime)	226
public static int cameraCommandEvent(String function, String action, String camera)	226
IRTrans	228
public static void IREvent(boolean received, int irtransid, String event)	228
Squeezebox	229
public static void SlimPowerEvent(int index, int power)	229
public static void SlimStatusEvent(int index, int status)	229
public static void SlimVolumeEvent(int index, int volume)	229
PBX	230
public static boolean PBXCallEvent(String host, String caller, String called)	230
Servizi di Rete	231
public static void LocationEvent(long MAC, InetAddress IP, int zoneld)	231
Metodi di Sistema	232
public static void StartupEvent()	232
public static String getUserVersion()	232
public static void TimeEvent(long time)	232
public static void programTimerEvent(String name)	232
public static String userCommand(String name, String param)	233

public static void DaylightEvent(boolean day)	234
public static int PowerEvent(int power)	234
public static void SunPositionEvent(int azimuth, int elevation)	234
public static String WebRootRequestEvent(InetAddress addr, boolean secure, String useragent)	235
static void SchedulerEvent(String groupname, String schedulename)	235
Metodi di Comando e Utilità	236
I/O Server	236
static void ioSet(String id, String value)	236
static String ioGet(String id)	236
DMX	237
static int dmxSet(int channel, int state)	237
static int dmxSet(int[] channels, int[] state)	237
static int dmxOff(int channel)	238
static int dmxOff(int[] channels)	238
static int dmxOn(int channel)	238
static int dmxOn(int[] channels)	239
static int dmxMerge(int from, int to, boolean merge)	239
static int dmxMerge(int[] channels, boolean merge)	240
static int dmxGet(int channel)	240
static void dmxOffTimerSet(int channel, int seconds)	240
static void dmxOffTimerClear(int channel)	241
static void dmxOffTimerPreset(int channel, int seconds)	241
static void dmxOffTimerReset(int channel, int seconds)	242
Telecamere	243
static void cameraRecTrigger(String camera, String source, int seconds)	243
static void cameraRecTriggerFull(String camera, String source, int seconds)	243
static int cameraCommand(String function, String action, String camera)	244
static void cameraMode(String camera, boolean enabled)	244
static void cameraRecMode(String camera, boolean enabled)	245

Porte di Comunicazione Seriale	246
static int writeComm(String portName, String data)	246
static String readComm(String portName, int len)	246
static int closeComm(String portName)	247
Plugins	248
static Bentel getBentelPlugIn(String id)	248
static ParadoxEVO getParadoxEVOPlugIn(String id)	248
static Tecnoalarm getTecnoalarmPlugIn(String id)	248
Modbus	250
static byte[] modbusReadCoils(String name, int unit, int address, int quantity)	250
static byte[] modbusReadDiscreteInputs(String name, int unit, int address, int quantity)	250
static byte[] modbusReadHoldingRegisters(String name, int unit, int address, int quantity)	251
static byte[] modbusReadInputRegisters(String name, int unit, int address, int quantity)	252
static byte[] modbusWriteSingleCoil(String name, int unit, int address, boolean value)	252
static byte[] modbusWriteSingleRegister(String name, int unit, int address, byte[] bytes)	253
static byte[] modbusWriteMultipleRegisters(String name, int unit, int address, byte[] bytes)	254
static byte[] modbusWriteMaskRegister(String name, int unit, int address, byte[] mask, byte[] bytes)	254
IRTrans	256
static int irtransCommand(String ird, String command)	256
Squeezebox	257
static String slimCommand(int index, String command)	257
static String slimCommand(String playerName, String command)	257
static int slimButton(int index, String button)	257
static int slimButton(String playerName, String button)	258

Public Announcement	259
static int audioPlay(String where, File file)	259
static int audioPlay(String where, String voice, String text)	260
Data Logger	261
static boolean dataLoggerRefresh(String name)	261
static boolean dataLoggerUpdate(String name, double value)	261
static boolean dataLoggerClear(String name)	261
static boolean dataLoggerSave(String type, String[] names, String path, boolean timestamp)	262
static boolean dataLoggerOptions(String name, String param, String value)	262
Leak Detector	263
static void leakDetectorClear(String name)	263
static void leakDetectorOptions(String name, String param, String value)	263
public static int leakDetectorUpdate(String name, double value)	264
Mail	265
static int sendMail(String to, String from, String subject, String body)	265
static int sendMail(String to, String from, String subject, Vector<String> body)	266
Log	268
static void messageLog(String message)	268
static void errorLog(String message)	268
Servizi di Rete	269
ping(String host, int timeout)	269
static int wakeOnLan(String broadcast, String address)	269
Metodi di Sistema	270
static boolean haActiveState()	270
static boolean isDaylight()	270
static long getNextSunrise(long now, boolean withoffset)	270
static long getNextSunset(long now, boolean withoffset)	270

static void programTimerSet(String name, int seconds)	271
static void programTimerClear(String name)	271
static void programTimerReset(String name, int seconds)	271
static void programTimerRepeat(String name, int seconds)	272
static void sleep(long millis)	272
static void schedulerRegister(String groupname, String schedulename, int interval)	272
static void SchedulerRemove(String groupname, String schedulename)	273
static void user(String name, String param)	273
static String varGet(String name)	273
static void varSet(String name, String value)	274
static String uiGet(String id, String attr)	274
static void uiSet(String id, String attr, String value)	274
static void powerSet(int power)	280

Appendici **281**

Data Logger	281
Data Logger di tipo Counter	282
Data Logger di tipo Range	286
Configurazione delle Telecamere	289
Configurazione Utenti	289
Acquisizione dei Frame	289
Motion Detection	290
Controllo PTZ	290
Configurazione di Server con Porte Seriali	291
IRTrans	291
Controller HW Group	291
Generico Gateway Seriale	292

Servizio di Localizzazione Wi-Fi	293
Servizi di Telefonia	294
Public Announcement	295
Riproduzione File Audio	295
Il Motore Text-to-Speech	295
eSpeak	296
Acapela	296
Mac OS X	296
Il Formato dei File di Log	298
Avvio di HSYCO Senza Chiave di Licenza	298
Avvio di HSYCO Normale	299
Errori di Connessione all'I/O Server	300
Errori di Connessione a IRTrans	300
Errori di Scrittura delle Immagini delle Telecamere	300
Messaggi di Sicurezza	301
Comandi Inviati dall'Interfaccia Web	302
Localizzazione dei Client Web Wi-Fi	303
Log delle Chiamate Telefoniche	303
Il File security.log	303
Lista di Controllo Accessi	304
Certificati SSL per la Crittografia	306
Database SQL	309
Connessione	310
Disconnessione	310
Tabelle	311

Operazioni sulle Tabelle	311
Query	312
Inserimento	313
Aggiornamento	313
Il File access.ini	314
Release Notes	317
3.1.2	317
3.1.1	318
3.1.0	319
3.0.3	321
3.0.2	322
3.0.1	323
3.0.0	324
2.10.4	328
2.10.3	328
2.10.2	329
2.10.1	329
2.10.0	329
2.9.0	333
2.8.4	337
2.8.3	338
2.8.2	338
2.8.1	339
2.8.0	340
2.7.1	343

2.7.0	344
2.6.1	346
2.6.0	346
2.5.4	348
2.5.3	349
2.5.2	350
2.5.1	351
2.5.0	351
2.4.3	352
2.4.2	352
2.4.1	352
2.4.0	352
2.3.0	353
2.2.2	353
2.2.1	353
2.2.0	353
2.1.7	354
2.1.6	354
2.1.5	354
2.0	355

Installazione

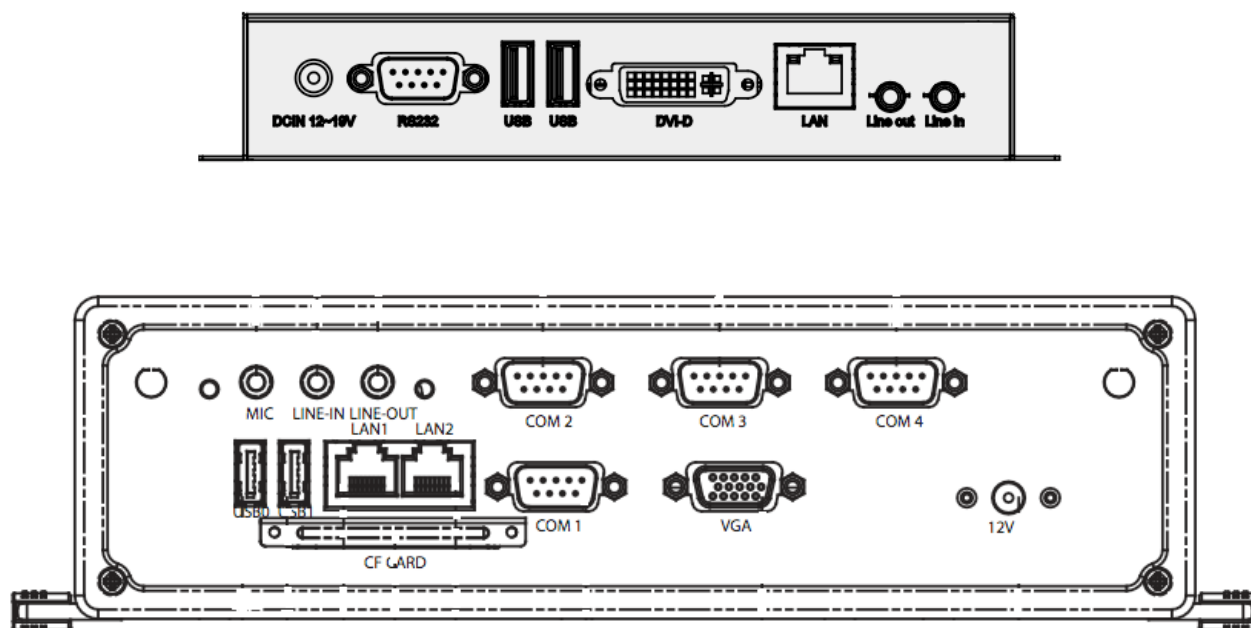
HSYCO SERVER viene fornito già installato con la versione corrente di HSYCO, su una distribuzione Linux Ubuntu 10.04 LTS Server privata di tutti i servizi non essenziali. E' inoltre installato il Java SE Runtime Environment versione 1.6.0. Non sono richiesti altri componenti software.



Questo capitolo descrive le modalità di accesso per la gestione di base del sistema operativo e la struttura dei file che costituiscono HSYCO.

Accesso al Sistema Operativo

HSYCO SERVER ha due distinte configurazioni hardware standard: una con un server di dimensioni compatte dotato di una porta Ethernet e una porta seriale, e una con un server più grande dotato di due porte Ethernet e quattro porte seriali.



HSYCO SERVER viene fornito già configurato con la seguente configurazione di rete:

- IP address: 192.168.0.50
- Net mask: 255.255.255.0
- Default Gateway: 192.168.0.1
- DNS: 192.168.0.1

Nel seguito di questo manuale l'indirizzo IP 192.168.0.50 sarà utilizzato in tutti gli esempi come indirizzo di HSYCO SERVER.

Sul server con due porte Ethernet, la porta Ethernet normalmente configurata è quella posizionata a fianco delle porte USB, indicata come **LAN1** nella figura seguente. La porta Ethernet **LAN2**, a destra, è funzionante ma non configurata.

La Console di Configurazione

HSYCO Server fornisce una console di manutenzione a basso livello, utilizzata per riavviare il sistema operativo, cancellare le impostazioni o ripristinare l'intero sistema alla configurazione di fabbrica. Questa console non viene utilizzata per normali attività.

La console è accessibile attraverso una tastiera e un monitor collegati all'uscita video VGA/DVI/HDMI di HSYCO Server.

Server Configuration Console

```
1) Reboot the system
2) Halt the system
3) Stop HSYCO Server
4) Start HSYCO Server
5) Restore network configuration to factory defaults (192.168.0.50)
6) Delete all PINs and file server password and restore default values
7) Restore database to factory defaults. All data will be lost
8) Restore system software to factory defaults. All data will be lost
Type a function number and press Enter [1, 2, 3, 4, 5, 6, 7, 8]:
```

Digitare il numero della funzione desiderata e Invio, dopo i beep emessi premere Y per confermare (il sistema emetterà un numero di beep pari al numero della funzione richiesta; è quindi possibile accedere alle funzioni della console utilizzando solo la tastiera e la notifica audio, senza necessariamente collegare un monitor).

La funzione 1) spegne e riavvia il sistema operativo.

La funzione 2) spegne il sistema operativo e lo ferma. Per farlo ripartire è necessario riavviarlo.

Le funzioni 3) e 4) vengono utilizzate per fermare temporaneamente e per avviare il processo di HSYCO Server.

Utilizzare le funzioni dalla 5) alla 8) con cautela. Tali funzioni cancelleranno le impostazioni di configurazione, i dati utente e le impostazioni di sicurezza del sistema. Inoltre consentono ad un utente in locale di accedere fisicamente al server e ai dati senza alcuna restrizione.

La funzione 5) ripristina la configurazione di rete di default. Questa funzione è utile nel caso l'utente abbia sbagliato a impostare i parametri di rete e non riesca più ad accedere al server tramite la connessione di rete. In questo modo è possibile cancellare la configurazione e accedere nuovamente al server.

La funzione 6) rimuove tutti i codici PIN e PUK impostati dall'utente, reimpostando il PIN di default 00000 con diritti di amministratore. Inoltre ripristina la password del sistema operativo a quella di default.

La funzione 7) elimina tutti i dati del database di HSYCO, incluse le variabili persistenti.

La funzione 8) esegue un completo reset alle impostazioni di fabbrica, eliminando tutti i dati, la configurazione e i programmi impostati dall'utente. Il sistema viene riportato alla versione e alle impostazioni di HSYCO iniziali.

Pagine di Amministrazione

HSYCO SERVER viene fornito già configurato con una pagina di amministrazione che permette l'accesso ad alcune funzioni di amministrazione, come il cambio delle impostazioni di rete, il cambio della password del sistema operativo e la gestione degli utenti di HSYCO.

Per accedere al menu di amministrazione, inserire il seguente URL nel browser Web:

<https://192.168.0.50/hsycoserver/admin>

Poiché il certificato SSL usato per proteggere la comunicazione tra il Browser Web e HSYCO SERVER è auto-generato, occorrerà accettare il certificato del server.



Su iPod touch, iPhone, o iPad premere il bottone Continua, sugli altri browser seguire le finestre di dialogo per accettare in maniera permanente il certificato.

A questo punto compariranno le pagine di autenticazione di PIN e PUK. Inserire PIN e PUK pre-configurati e premere OK:

PIN: 0000

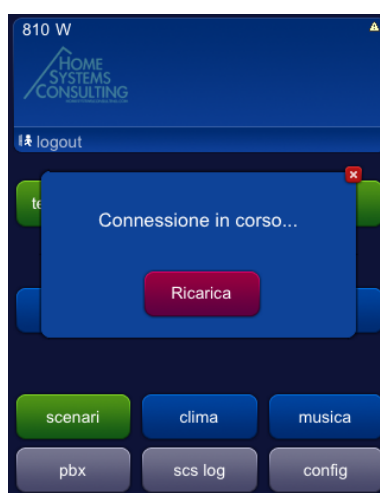
PUK: 00000 0000 00000



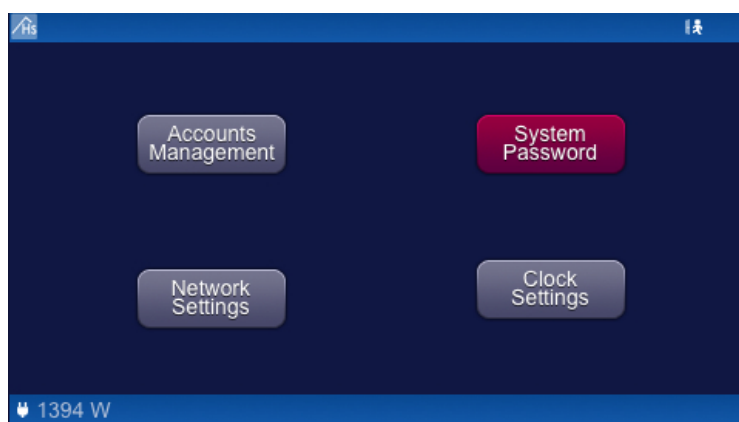
I PIN e PUK pre-configurati di default sono gli stessi per ogni HSYCO SERVER, in modo da permettere un facile accesso all'interfaccia utente durante la fase iniziale di installazione. Sia PIN che PUK devono essere perciò sostituiti immediatamente da codici diversi, composti da una serie complessa di numeri.

Se appare un triangolo giallo nella parte alta del display dell'interfaccia significa che il browser non riesce a comunicare con il server HSYCO. Uno dei possibili motivi può essere causato dalla perdita del certificato del server.

Per rigenerare il certificato premere sul triangolo giallo o eseguire il refresh della pagina Web premendo il bottone Ricarica

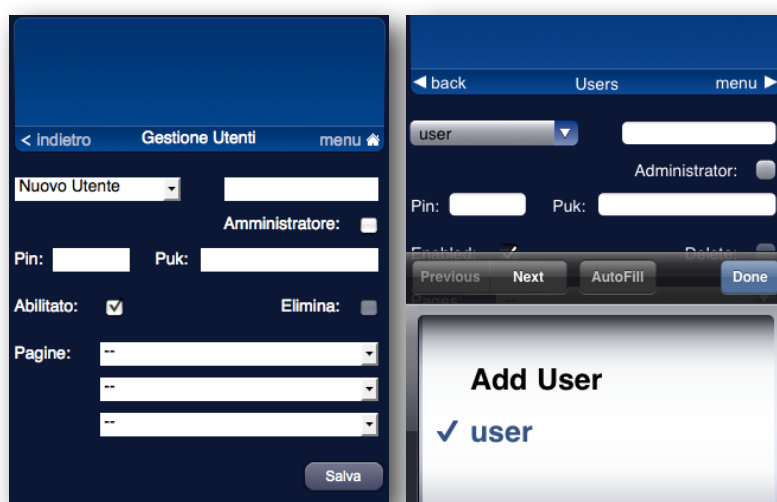


La pagina di amministrazione standard dispone di quattro tasti. In seguito verrà spiegato come riconfigurare la pagina di amministrazione e le altre pagine di HSYCO, a seconda delle esigenze dei diversi utenti.



Gestione degli Account

La pagina di gestione degli account permette di gestire in maniera completa i singoli utenti dall'interfaccia web di HSYCO. E' possibile impostare e revocare i diritti di amministratore, abilitare e disabilitare un utente, modificare PIN e PUK, eliminare un utente, ed infine definire i diritti di accesso alle sotto-directory. Se non viene specificata alcuna pagina attraverso l'interfaccia Web, l'utente non ha limiti e può accedere a tutte le pagine presenti.



Per cambiare PIN e PUK predefiniti, selezionare **Utente**, inserire i nuovi codici PIN e PUK nei relativi campi, e premere il bottone **Salva** per confermare.

E' possibile inoltre creare uno o più nuovi utenti, dei quali almeno uno con diritti di amministratore, effettuare nuovamente il login con i nuovi PIN e PUK dell'amministratore ed eliminare l'utente pre-configurato.

Password di Sistema

La password di sistema è usata per l'accesso alla console del sistema operativo, direttamente o con SSH, e ai file di HSYCO attraverso il servizio di rete di cartelle condivise. Questa password è necessaria per l'accesso a HSYCO SERVER per la manutenzione e per il cambio della configurazione.

Per cambiare la password di sistema, effettuare il login con il PIN e PUK corrispondenti all'utente con privilegi di amministratore.



Cliccare sul bottone System Password nella pagina di amministrazione e inserire la nuova password due volte, infine premere il bottone **Salva** per confermare.

Per garantire un'adeguata sicurezza la password scelta deve essere composta da lettere e numeri, non deve contenere parole conosciute e dovrebbe essere composta da almeno 10-16 caratteri.

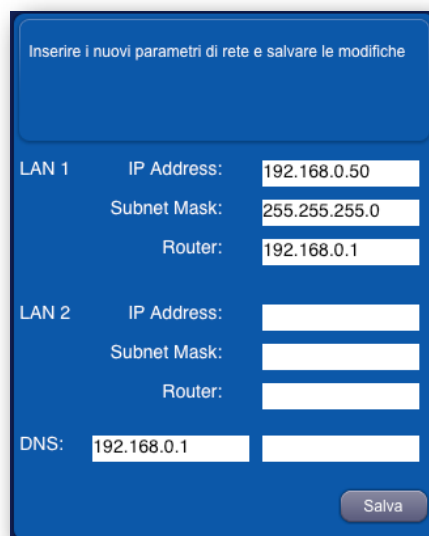
La password dovrebbe essere conservata in un posto sicuro. In caso venga dimenticata, ma si continui ad avere accesso a HSYCO, si può semplicemente accedere alla pagina della password e digitare una nuova password.

La password di sistema dovrebbe essere conosciuta solo dal proprietario del sistema. Per consentire un accesso per la manutenzione di HSYCO ad altri, si consiglia di cambiare la password con una nuova password provvisoria, e ritornare in seguito alla propria password segreta.

Occorre ricordare che la password di sistema garantisce libero accesso alla configurazione del sistema operativo, alla configurazione di HSYCO, inclusa la gestione degli utenti, e a tutti i file, come le immagini registrate da tutte le telecamere e i file di log.

Impostazioni di Rete

La pagina delle impostazioni di rete permette di modificare i parametri di rete necessari per il funzionamento di HSYCO. Errori nell'impostazione di questi parametri possono rendere HSYCO SERVER inaccessibile dalla rete e attraverso l'interfaccia web di HSYCO.

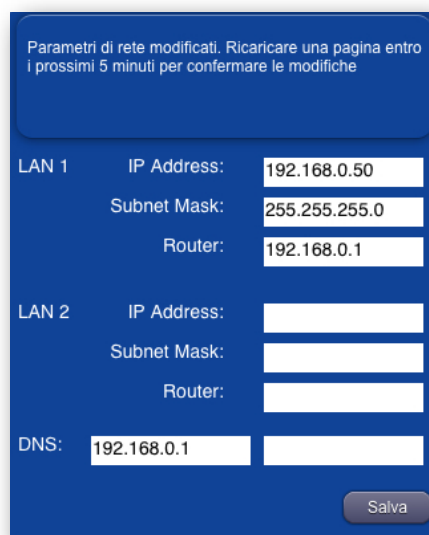


Inserire i nuovi parametri di rete e salvare le modifiche

LAN 1	IP Address:	192.168.0.50
	Subnet Mask:	255.255.255.0
	Router:	192.168.0.1
LAN 2	IP Address:	
	Subnet Mask:	
	Router:	
DNS:	192.168.0.1	

Salva

HSYCO implementa un meccanismo di sicurezza che nella maggior parte dei casi ristabilisce, dopo 5 minuti dalla modifica apportata, le impostazioni di rete della configurazione originale nel caso in cui l'accesso di rete a HSYCO sia stato perso dopo un cambiamento.



Parametri di rete modificati. Ricaricare una pagina entro i prossimi 5 minuti per confermare le modifiche

LAN 1	IP Address:	192.168.0.50
	Subnet Mask:	255.255.255.0
	Router:	192.168.0.1
LAN 2	IP Address:	
	Subnet Mask:	
	Router:	
DNS:	192.168.0.1	

Salva

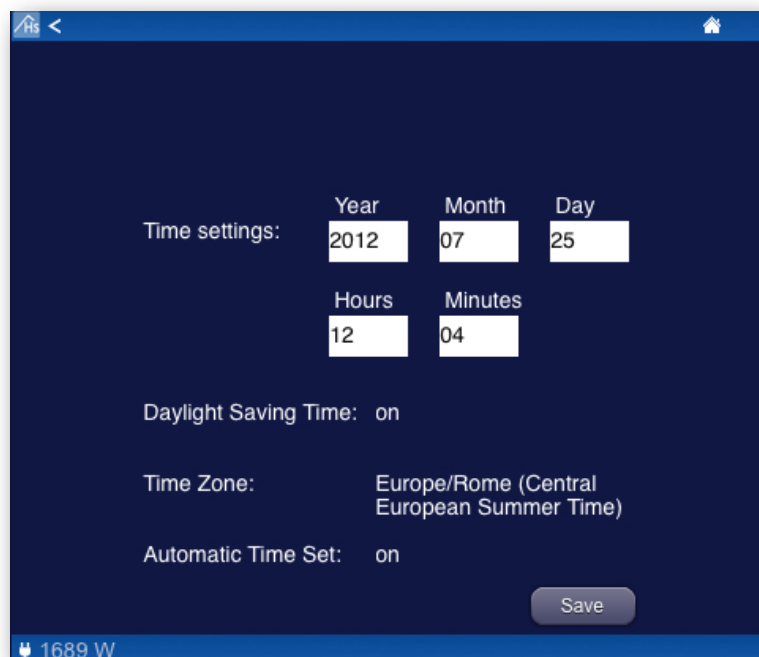
Per modificare le impostazioni di rete, inserire i nuovi parametri e premere il bottone **Salva**.

Dopo la modifica, si hanno a disposizione 5 minuti per effettuare un nuovo accesso ed entrare nella pagina delle impostazioni di rete per confermare le modifiche. Non occorre premere ancora il bottone Save, accedendo alla pagina verrà comunicato a HSYCO che disponete ancora di accesso alla rete, e le nuove impostazioni di rete saranno registrate in maniera permanente.

La pagina delle impostazioni di rete non è normalmente necessaria dopo la prima installazione, ed è di solito cancellata dal menu per evitare modifiche accidentali alla configurazione di rete.

Impostazioni dell'orologio

La pagina Clock Settings imposta l'orologio del server e il fuso orario, e abilita l'aggiornamento automatico dell'ora basato sull'ora dei server da Internet.



Time settings:

Year	Month	Day
2012	07	25

Hours	Minutes
12	04

Daylight Saving Time: on

Time Zone: Europe/Rome (Central European Summer Time)

Automatic Time Set: on

Save

1689 W

Per modificare le impostazioni di data e ora, inserire i nuovi parametri e premere il bottone **Salva**.

Tutti gli altri parametri di questa pagina sono di sola lettura ed è possibile modificare le impostazioni dal file hsyco.ini.

Time Zone viene utilizzato per ottenere il fuso orario locale e regolare automaticamente l'ora in base all'ora legale.

Se Automatic Time è abilitato (on), HSYCO imposta automaticamente ora/data locali in base ai dati provenienti dai server Network Time Protocol (NTP).

Il Manager

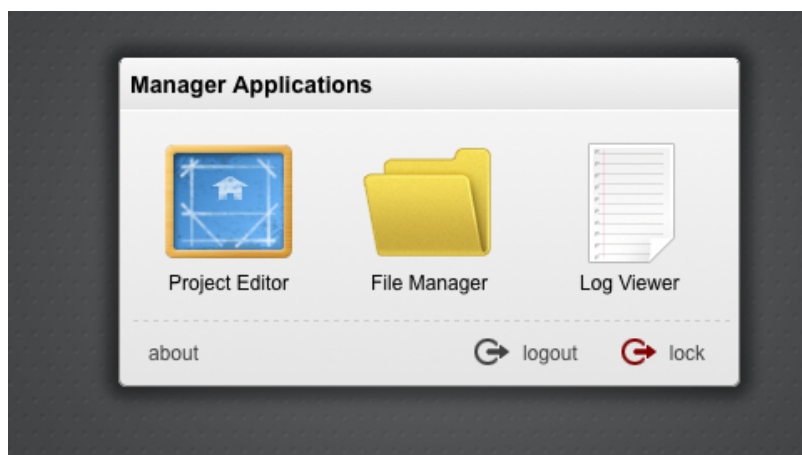
Il Manager è uno strumento applicativo Web-based che consente all'utente di creare interfacce utente grafiche personalizzate e accedere a tutti i file di configurazione del sistema.

Il Manager è accessibile da qualsiasi dispositivo dotato di un Web browser moderno, inclusi tablet come iPad di Apple o dispositivi più piccoli come iPhone o iPod touch, anche se è consigliabile una dimensione dello schermo più grande per questo tipo di utilizzo. La versione di Microsoft Internet Explorer 8 e quelle precedenti non sono supportate e di IE9 non è garantito il funzionamento.

Per accedere al Manager, digitare il seguente URL nel browser Web:

`https://192.168.0.50/hsycoserver/manager`


Per utilizzare il Manager è necessario che l'utente si autentichi utilizzando un PIN con diritti di amministratore.

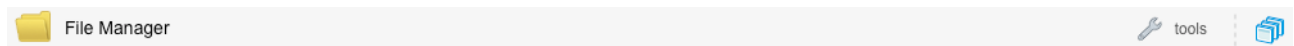


All'interno del Manager sono presenti tre applicazioni: il Project Editor, l'editor grafico con cui l'utente può personalizzare l'interfaccia utente; il File Manager, che fornisce l'accesso a tutti i file di HSYCO (come il file manager standard presente su un computer), totalmente Web-based e in grado di lavorare

direttamente sui file di HSYCO Server; il Log Viewer, che consente di visualizzare in tempo reale i log di HSYCO.

Toccando “about” nell’angolo in basso a sinistra, viene visualizzato un pop-up con le informazioni sulla versione di HSYCO installata.

E’ possibile spostarsi da un’applicazione a un’altra premendo l’icona  sulla barra in alto.

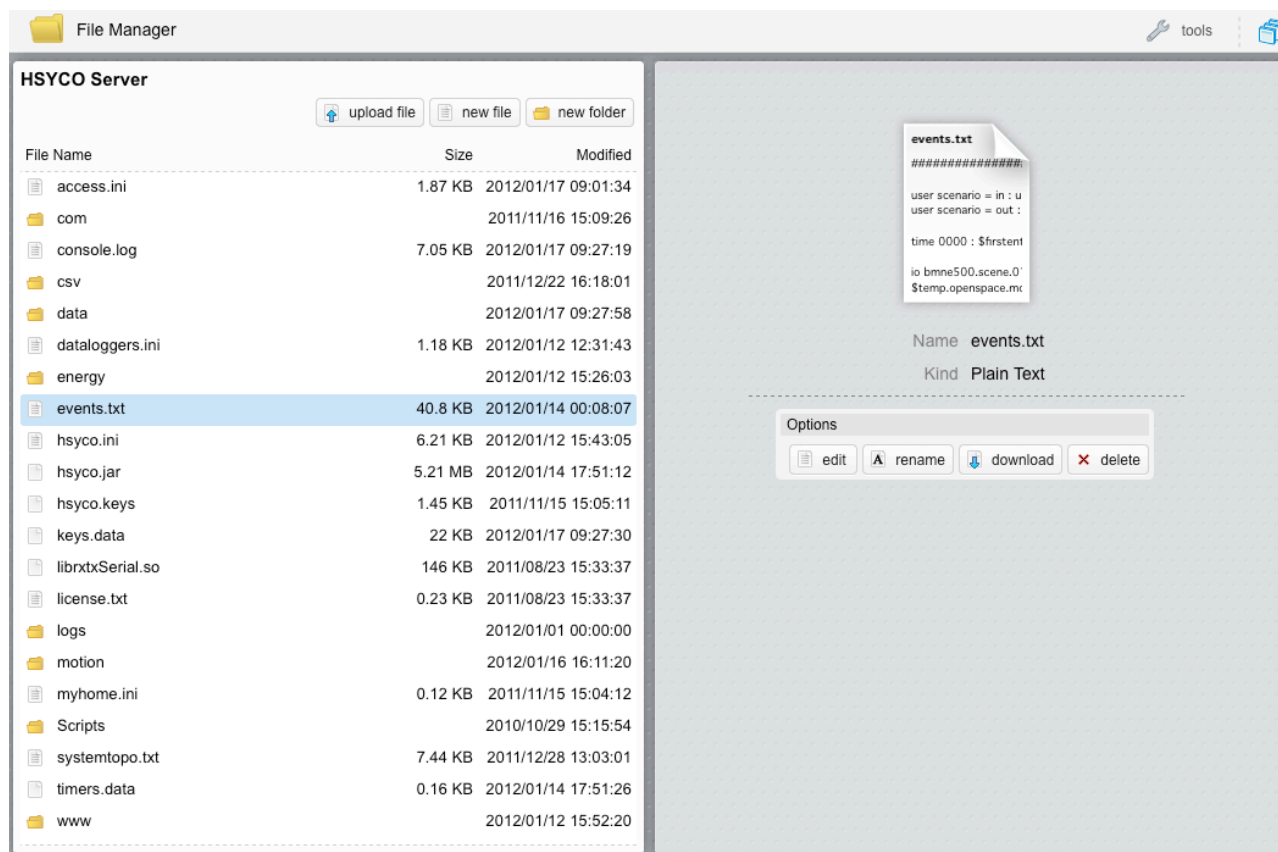


Utilizzando un PC con tastiera, è possibile passare velocemente da un’applicazione alle altre premendo contemporaneamente i tasti [Ctrl] e [Shift]. Verrà visualizzato un pannello di selezione delle applicazioni, che rimarrà visibile fino al rilascio del tasto [Ctrl]. Premere ripetutamente il tasto [Shift] per selezionare l’applicazione desiderata, quindi rilasciare il tasto [Ctrl].

Il Project Editor sarà discusso più avanti, nel capitolo Personalizzazione.

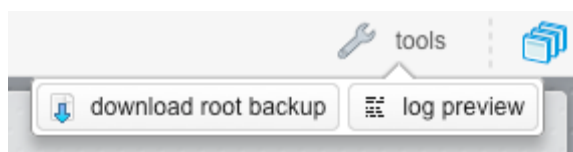
Il File Manager

Il File Manager offre tutti gli strumenti standard convenzionali utilizzati per gestire i file, incluso un editor di testo per modificare i file testuali senza la necessità di doverli scaricare sul computer.

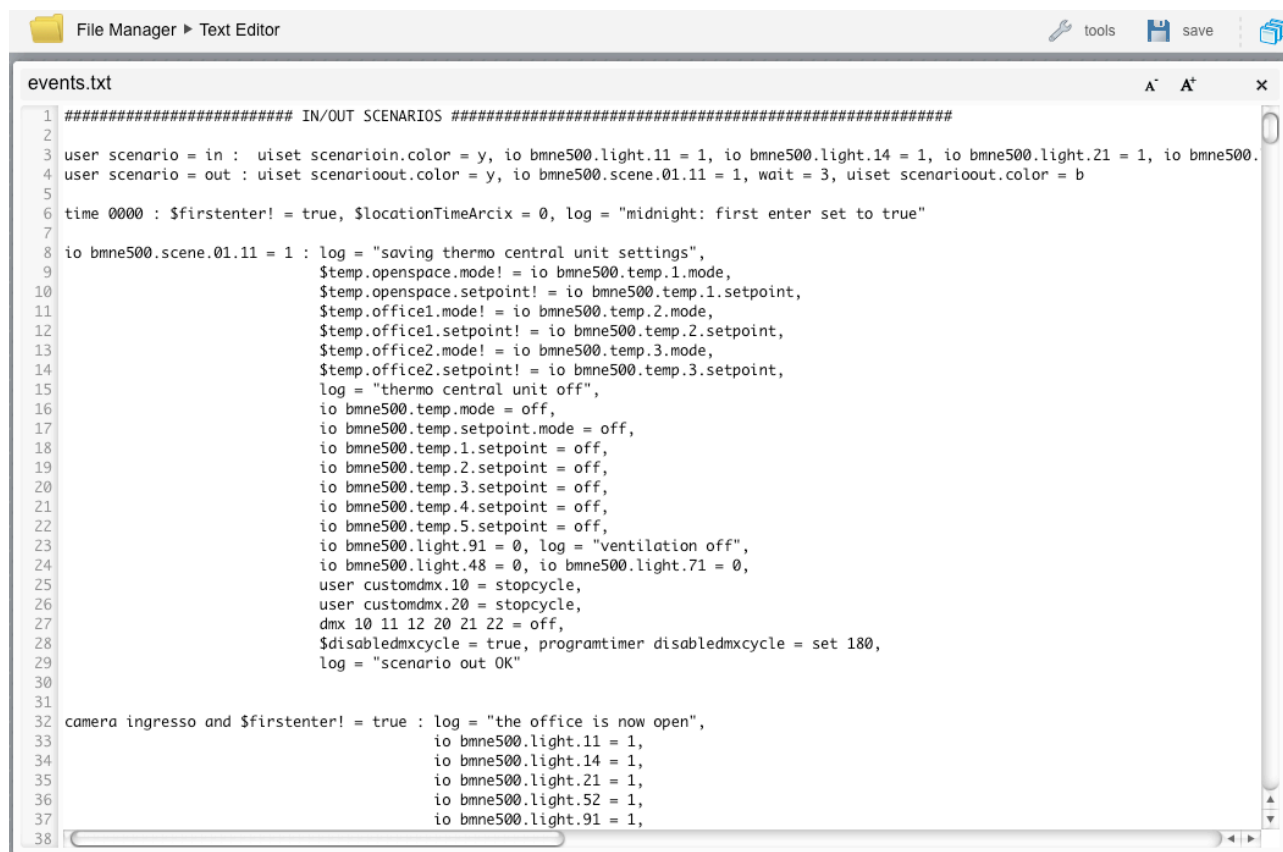


E' possibile effettuare il download e l'upload di file singoli o di directory in formato ZIP, creare, rinominare e eliminare file e directory.

Inoltre, attraverso lo strumento download root backup, è possibile eseguire il backup completo di tutti i file di configurazione (ad eccezione delle directory logs e motion). Questa operazione può richiedere molto tempo per generare e scaricare il file backup.zip.



E' anche possibile attivare la visualizzazione di una piccola finestra di log in tempo di reale, utile soprattutto quando si stanno modificando i file e si vuole controllare il log per eventuali correzioni.

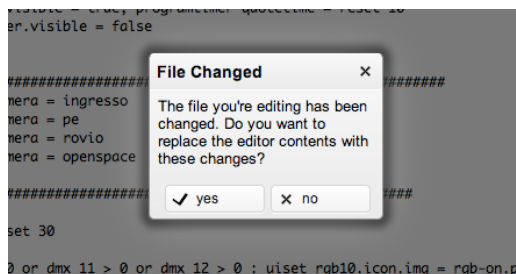


```

1 ##### IN/OUT SCENARIOS #####
2
3 user scenario = in : uiset scenarioin.color = y, io bmne500.light.11 = 1, io bmne500.light.14 = 1, io bmne500.light.21 = 1, io bmne500.
4 user scenario = out : uiset scenariout.color = y, io bmne500.scene.01.11 = 1, wait = 3, uiset scenariout.color = b
5
6 time 0000 : $firstenter! = true, $locationTimeArcix = 0, log = "midnight: first enter set to true"
7
8 io bmne500.scene.01.11 = 1 : log = "saving thermo central unit settings",
9                             $temp.openspace.mode! = io bmne500.temp.1.mode,
10                            $temp.openspace.setpoint! = io bmne500.temp.1.setpoint,
11                            $temp.office1.mode! = io bmne500.temp.2.mode,
12                            $temp.office1.setpoint! = io bmne500.temp.2.setpoint,
13                            $temp.office2.mode! = io bmne500.temp.3.mode,
14                            $temp.office2.setpoint! = io bmne500.temp.3.setpoint,
15                            log = "thermo central unit off",
16                            io bmne500.temp.mode = off,
17                            io bmne500.temp.setpoint.mode = off,
18                            io bmne500.temp.1.setpoint = off,
19                            io bmne500.temp.2.setpoint = off,
20                            io bmne500.temp.3.setpoint = off,
21                            io bmne500.temp.4.setpoint = off,
22                            io bmne500.temp.5.setpoint = off,
23                            io bmne500.light.91 = 0, log = "ventilation off",
24                            io bmne500.light.48 = 0, io bmne500.light.71 = 0,
25                            user customdmx.10 = stopcycle,
26                            user customdmx.20 = stopcycle,
27                            dmx 10 11 12 20 21 22 = off,
28                            $disabledmxcycle = true, programtimer disabledmxcycle = set 180,
29                            log = "scenario out OK"
30
31
32 camera ingresso and $firstenter! = true : log = "the office is now open",
33                                           io bmne500.light.11 = 1,
34                                           io bmne500.light.14 = 1,
35                                           io bmne500.light.21 = 1,
36                                           io bmne500.light.52 = 1,
37                                           io bmne500.light.91 = 1,
38

```

All'interno dell'editor di testo, se il file che si sta attualmente editando viene modificato, comparirà un messaggio che avvisa l'utente che il file è stato modificato e chiede se si vuole caricare il file attuale con le nuove modifiche apportate o ignorare i cambiamenti.



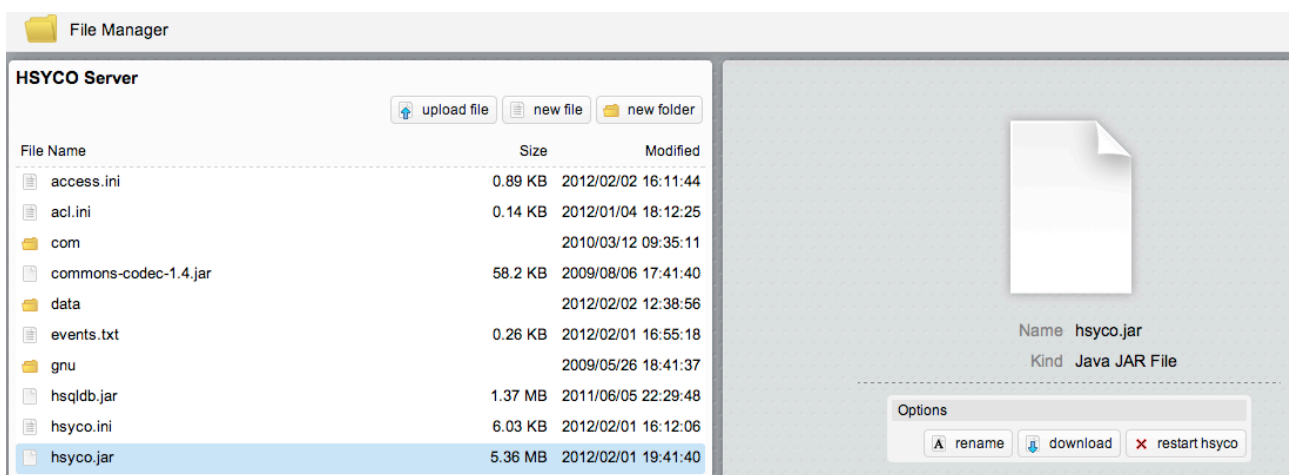
Se si accetta di caricare la nuova versione del file le modifiche effettuate fino a quel momento verranno perse.

Aggiornamento e riavvio di HSYCO

Per aggiornare la versione software di HSYCO è sufficiente usare il File Manager per copiare la nuova versione del file hsyco.jar nella root directory del server.

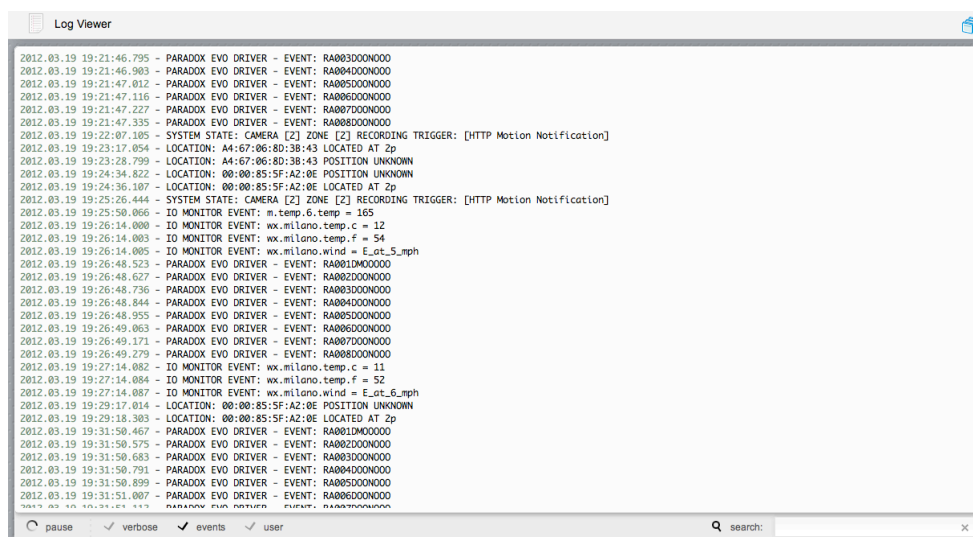
Completato il caricamento del file, il server si riavvierà ed il Manager sarà ricaricato dal browser Web.

E' anche possibile forzare manualmente il riavvio di HSYCO, selezionando il file hsyco.jar e toccando il tasto "restart hsyco" nel pannello di destra.




Il Log Viewer




Il Log Viewer visualizza in tempo reale le ultime 1000 righe del file di log giornaliero.




La barra degli strumenti in basso consente di filtrare le righe di log per la ricerca di una stringa di testo, o per escludere una stringa all'interno del log. Inserendo più parole verranno visualizzate o escluse tutte le righe che contengono almeno una delle parole.

Premere  **pause** per interrompere temporaneamente la visualizzazione delle righe di log. Premere nuovamente per riabilitare la scrittura. Le righe di log create durante la pausa non saranno più visualizzate in questa finestra, ma saranno comunque scritte nel file di log.

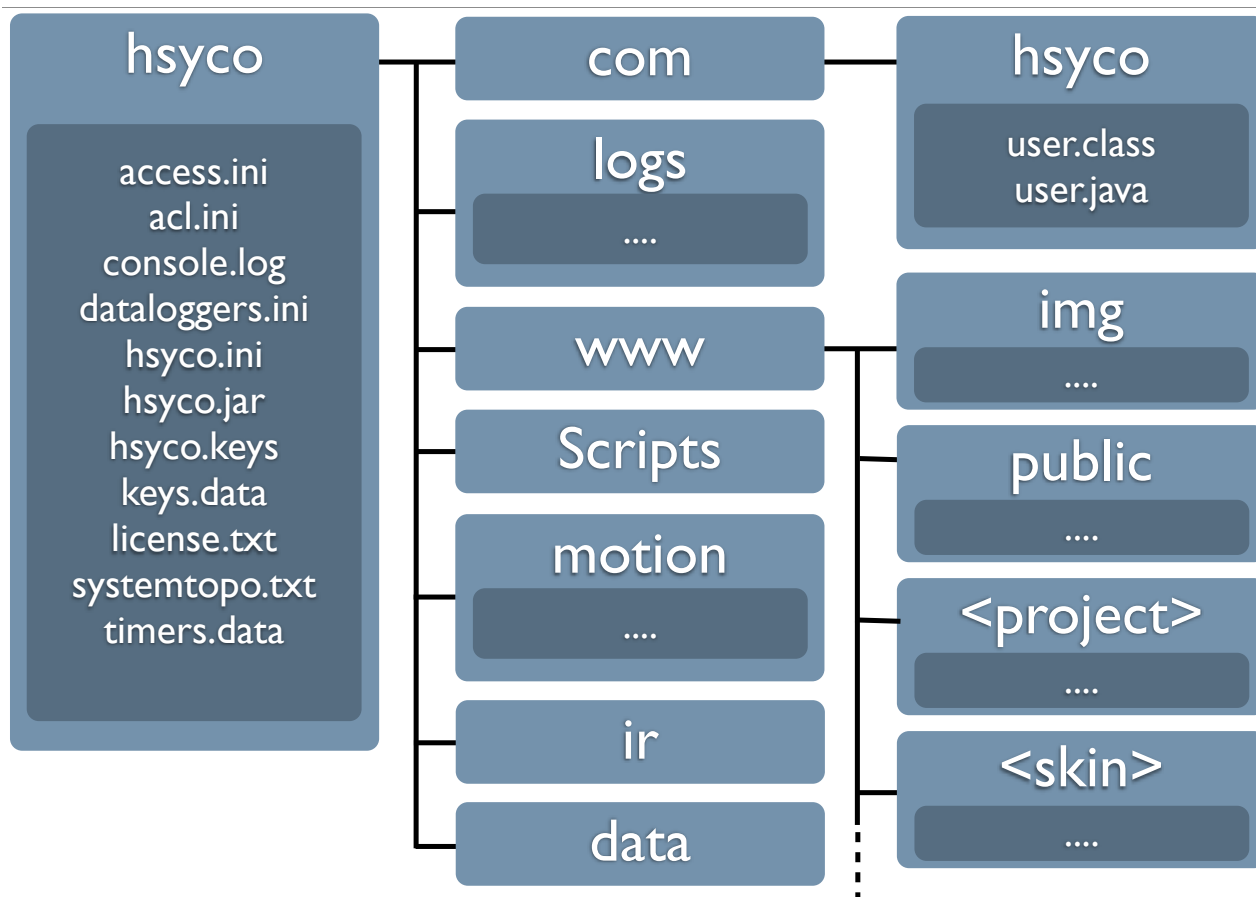
E' anche possibile selezionare il tipo di livello di log, senza apportare modifiche al file di configurazione principale hsyco.ini.

Premere  **verbose**,  **events** o  **user** per abilitare o disabilitare i livelli di log. La modifica dei livelli di log implica che tutti gli altri utenti che stanno consultando il log in quel momento visualizzeranno il cambiamento, così come tutto quello che viene effettivamente scritto nel file di log.

Premere  **search:** per cambiare tra la modalità di ricerca e la modalità di esclusione di una stringa.

Organizzazione dei File

La struttura generale della directory principale di HSYCO è rappresentata nella figura seguente.



hsyco/access.ini

Questo file contiene le informazioni relative ai **PIN/PUK** di accesso degli utenti. Il file può essere modificato manualmente per creare o cancellare PIN, ma normalmente viene modificato attraverso le pagine di gestione utenti nell'interfaccia Web di HSYCO. Viene anche aggiornato da HSYCO in seguito ad ogni accesso, con l'orario e l'indirizzo IP dell'ultimo accesso ed il numero di errori di accesso per ciascun utente.

hsyco/acl.ini

Questo file è utilizzato per definire liste di controllo accessi lato server per comandi utente Web-based. Il file è opzionale e definisce le regole per accettare o rifiutare comandi.

Per ulteriori dettagli consultare il capitolo Lista Controllo Accessi nella sezione Appendice.

hsyco/console.log

console.log è un file di log. Contiene informazioni ed eventuali messaggi di errore generati dal sistema operativo o dalla Java Virtual Machine.

Ad esempio, un messaggio come il seguente:

```
Exception in thread "main" java.lang.NoClassDefFoundError: com/
hsyco/OpenWebNetMonitor
at com.hsyco.hsyco.main(hsyco.java:4794)
Caused by: java.lang.ClassNotFoundException:
com.hsyco.OpenWebNetMonitor
```

indica che una delle librerie Java di HSYCO è mancante. Questo file è quindi utile soprattutto in fase di test o durante lo sviluppo di componenti Java aggiuntive, mentre non fornisce informazioni rilevanti per il funzionamento normale del sistema.

hsyco/hsyco.ini

E' il file principale di configurazione, che contiene tutti i parametri di configurazione generale di HSYCO. Viene descritto nel capitolo Configurazione. La modifica di questo file provoca dopo alcuni secondi il riavvio automatico di HSYCO.

hsyco/hsyco.jar

E' il file che contiene tutte le componenti del software HSYCO. L'aggiornamento di HSYCO ad una versione successiva comporta normalmente la sostituzione di questo singolo file, al cui interno è contenuto in formato compresso tutto il codice e le risorse del software. La modifica di questo file provoca dopo alcuni secondi il riavvio automatico di HSYCO.

hsyco/hsyco.keys

Questo file contiene il certificato **SSL** per la crittografia del traffico Web **HTTPS**. Quando HSYCO viene avviato, se questo file non è presente, viene generato automaticamente un nuovo certificato SSL in base al nome definito in hsyco.ini. Altrimenti HSYCO si limita a utilizzare il certificato contenuto in questo file, che può quindi anche essere stato generato da una **Certification Authority (CA)** ufficiale¹. A meno che non si sia scelto di utilizzare un certificato ufficiale, questo file viene quindi creato e gestito da HSYCO senza bisogno di interventi manuali.

hsyco/keys.data

E' il database delle chiavi di autenticazione sicura tra Web Browser ed HSYCO. Questo file viene creato e gestito da HSYCO e non deve essere modificato o cancellato.

¹ Come ad esempio Verisign o thawte.

hsyco/license.txt

Contiene la chiave di licenza software necessaria per il funzionamento di HSYCO. Quando HSYCO viene avviato per la prima volta, o se questo file viene rimosso prima dell'avvio, HSYCO crea un nuovo file contenente un codice crittografato che identifica univocamente il computer sul quale il software viene eseguito. Per consentire la corretta esecuzione di HSYCO è necessario che questo file contenga anche la chiave di licenza software, rilasciata da Home Systems Consulting, specifica per ogni computer.

Se la chiave di licenza non è presente nel file o non è corretta, HSYCO disabilita tutte le funzioni di accesso ai sistemi di campo, mantenendo in funzione solo il Web Server. Sarà quindi possibile accedere alle pagine configurate, ma non sarà possibile eseguire alcuna azione sugli impianti.

hsyco/systemtopo.txt

Contiene l'elenco completo di tutti i dispositivi presenti nell'impianto e i nomi delle zone **Wi-Fi**².

Il formato di questo file viene descritto dettagliatamente nel capitolo Configurazione.

systemtopo.txt può essere modificato in qualunque momento; le modifiche vengono automaticamente applicate entro pochi secondi senza che sia necessario il riavvio di HSYCO. Le pagine Web vengono ricaricate automaticamente in seguito ad ogni modifica di questo file.

hsyco/timers.data

E' il database delle impostazioni dei **timer**, quindi ad esempio sveglie, irrigazione o temporizzazioni luci, gestiti da HSYCO. Questo file viene creato e gestito da HSYCO e non deve essere modificato o cancellato.

² Cioè l'elenco dei diversi Access Point che costituiscono la rete wireless locale.

hsyco/com/hsyco

Questa directory contiene i file Java personalizzabili. In particolare deve essere sempre presente il file ***user.class***, anche solo nella sua versione originale nel caso non siano state fatte personalizzazioni. Il corrispondente file sorgente ***user.java*** non è necessario che sia presente per il corretto funzionamento del software.

hsyco/www

La directory `www` contiene tutti i file personalizzabili relativi alle pagine Web dell'interfaccia utente. In realtà la maggior parte dei file necessari sono inclusi nel package ***hsyco.jar*** e quindi non sono presenti sotto `www`; tuttavia è possibile inserire nelle posizioni corrispondenti in `www` nuovi file personalizzati, che saranno utilizzati con precedenza rispetto ai file standard contenuti in `hsyco.jar`.

Quando HSYCO rileva qualunque modifica ai file presenti in `www` e nelle sotto-directory, forza automaticamente il caricamento della pagina Web sui browser collegati, in modo da rendere immediatamente visibili le modifiche effettuate.

Di seguito vengono descritte le directory e i file standard, tuttavia la struttura delle directory e dei file può essere modificata in base alle esigenze di personalizzazione.

hsyco/www/<project>/index.hsm

La directory `<project>`, il cui nome può essere liberamente modificato, contiene i file di descrizione delle pagine di navigazione. All'interno di ogni directory `<project>` deve essere presente il file ***index.hsm*** che contiene la descrizione completa dell'interfaccia Web. Il formato di questo file è descritto estesamente nel capitolo Personalizzazione.

Per questa directory non è possibile utilizzare il nome *img* ed i nomi degli skin standard o personalizzati, che sono riservati. Non è inoltre possibile avere più file di descrizione dell'interfaccia Web all'interno della stessa directory.

hsyco/www/<project>/img

Contiene i file delle immagini associate, nel file `systemtopo.txt`, a ciascun elemento dell'impianto di illuminazione ed automazione o qualunque altra immagine utilizzata nelle pagine Web.

I file contenuti in questa sotto-directory hanno la precedenza rispetto ai file contenuti della directory `hsyco/www/img`.

hsyco/www/<skin>

Possono essere creati skin personalizzati, salvati come sotto-directory di `www`. I file degli skin standard sono già inclusi nel file `hsyco.jar`. E' possibile creare una directory con lo stesso nome di uno skin standard, ma è importante tenere presente che i file della directory contenuta in `www` sostituiscono i file dello skin standard.

hsyco/www/img

Contiene i file delle immagini associate, nel file `systemtopo.txt`, a ciascun elemento dell'impianto di illuminazione ed automazione o qualunque altra immagine utilizzata nelle pagine Web. I file contenuti nella directory `hsyco/www/<project>/img` hanno la precedenza rispetto ai file contenuti in questa directory.

hsyco/www/public

Il parametro `HTTPServerPublicDirectory` definito nel file `hsyco.ini` consente di abilitare la funzione di Web Server pubblico, per servire, senza autenticazione, i

file presenti nella directory specificata. Il nome di questa directory può essere anche diverso da “public”. Il Web Server pubblico risponde solo ai client che appartengono alla rete di indirizzi IP sicuri, via HTTP e HTTPS.

hsyco/motion

Contiene le immagini dei singoli frame (fotogrammi) registrati da ciascuna telecamera a seguito di eventi di motion detection. La directory motion contiene a sua volta una sotto-directory per ciascuna telecamera, con gli stessi nomi delle telecamere come definiti in hsyco.ini. All'interno di queste directory i file sono ulteriormente organizzati in altri due livelli di sotto-directory. La gestione di queste directory e dei file è a carico di HSYCO.

Per evitare malfunzionamenti nella riproduzione e potenzialmente nella registrazione degli eventi si raccomanda di non cancellare parzialmente i file. Qualora necessario è possibile cancellare tutti i file di una telecamera e successivamente far ripartire HSYCO in modo che gli indici delle immagini vengano aggiornati correttamente.

HSYCO provvede automaticamente a cancellare i file e le relative sotto-directory trascorso il tempo di conservazione delle immagini, impostato in hsyco.ini per ciascuna telecamera.

Inoltre i frame più vecchi vengono cancellati automaticamente anche prima del tempo limite di conservazione, nel caso in cui lo spazio disponibile su disco scenda al di sotto di un valore minimo di sicurezza.

hsyco/logs

Contiene i file di log generati da HSYCO. Il dettaglio delle informazioni riportate nei file di log dipende dalle impostazioni della modalità di log definite in hsyco.ini. Normalmente sono comunque tracciati tutti gli eventi rilevanti rilevati

sul campo, i comandi inviati, le informazioni di accesso degli utenti ed eventuali errori.

Alla partenza di HSYCO vengono inoltre riportate le informazioni relative alla esecuzione dei diversi moduli di gestione di HSYCO. Alcuni dei più importanti messaggi di log sono riportati in Appendice.

HSYCO scrive un file di log distinto per ogni giorno, all'interno di una directory per ogni anno, nel formato **MMGG-message.log**. Ad esempio il file di log del giorno 12 Aprile 2012 sarà `hsyco/logs/2012/0412-message.log`.

I file di log possono essere rimossi in qualunque momento senza causare problemi. HSYCO non prevede una procedura automatica di cancellazione dei file di log.

hsyco/ir

Questa directory può essere vuota o non essere presente. Contiene i file di comandi IR codificati in formato CCF, utilizzabili per inviare comandi IR utilizzando gli emettitori IRTrans.

I file devono avere estensione `.ccfhex`. Il nome di ogni file corrisponde al nome del database che raccoglie un insieme di comandi distinti. Ciascun file contiene normalmente più comandi, nel formato:

command_name = hex_ccf_data

La stringa CCF di ciascun comando può essere spezzata su più righe, e deve essere scritta in formato esadecimale.

hsyco/dataloggers.ini

Questo file contiene la dichiarazione di fasce orarie per i data logger di tipo counter (contatore). Ogni riga imposta le regole per la fascia oraria di un data logger e il formato da utilizzare è il seguente:

datalogger_name; slot_id; time_range; days_of_week; date; rate

Quando viene aggiornato un data logger di tipo counter elencato in questo file, le regole vengono applicate dall'alto al basso per stabilire a quale fascia oraria appartiene il nuovo valore e eventualmente quale tariffa applicare.

La seguente tabella descrive ciascun campo di una riga:

Campo	Formato	Descrizione
<i>datalogger_name</i>	stringa	specifica a quale data logger la fascia oraria viene applicata
<i>slot_id</i>	numero intero positivo	identificativo per la fascia oraria. Più righe possono riferirsi allo stesso identificativo di fascia
<i>time_range</i>	hh:mm-hh:mm	specifica l'intervallo di tempo in cui la fascia oraria si applica. L'asterisco (*) indica che la regola si applica a tutti gli intervalli di tempo.
<i>days_of_week</i>	lista di giorni della settimana	specifica i giorni della settimana indicati in questa regola. I giorni vengono identificati da una cifra compresa tra 1 (Lunedì) e 7 (Domenica). Per esempio, scrivendo "135" all'interno di questo campo la regola si applica ai giorni Lunedì (1), Mercoledì (3), e Venerdì (5). L'asterisco (*) indica che la regola si applica a tutti i giorni della settimana.
<i>date</i>	yyyy/mm/dd o yyyy/mm/dd-yyyy/mm/dd	specifica la data o l'intervallo di date a cui la fascia oraria si applica. Nel caso di intervallo (yyyy/mm/dd-yyyy/mm/dd) alla data di inizio o quella di fine può essere riportato un asterisco che indica che la regola verrà applicata rispettivamente fino o dalla data specificata. Per esempio, l'intervallo "*-2000/12/31" include tutte le date che precedono e includono il 31 Dicembre 2000. Se viene utilizzato il formato data esatto (yyyy/mm/dd), allora a ogni campo può essere riportato l'asterisco per indicare che la regola si applica per tutti gli anni, o tutti i mesi, o tutti i giorni. Ad esempio, "* /12/25", indica che la regola viene applicata il 25 Dicembre di ogni anno. L'asterisco (*) indica che la regola si applica a tutti i giorni di ogni anno
<i>rate</i>	numero float positivo	specifica la tariffa (un fattore moltiplicativo) da applicare alla fascia oraria.

hsyco/data

HSYCO integra HSQLDB (HyperSQL Database), un sistema per la gestione di basi di dati relazionali al quale le applicazioni possono collegarsi tramite il relativo driver JDBC API.

L'HSQLDB è direttamente integrato all'interno del core di HSYCO. Non è necessario aggiungere alcun file jar al classpath di HSYCO.

Il database utilizzato da HSYCO si chiama "hsyco" e i file sono salvati nella directory data/. La directory data/ è da considerarsi come un percorso riservato ed è pertanto sconsigliabile utilizzarla per memorizzare database dell'utente o, in generale, per altri scopi.

hsyco/Scripts

Questa directory può essere vuota o non essere presente. Quando presente contiene una copia degli script di sistema operativo necessari per l'esecuzione automatica di HSYCO all'avviamento del server³.

In questa directory è inoltre possibile scrivere ulteriori script eventualmente richiamati dal codice personalizzato in user.java.

³ HSYCO SERVER è basato sulla distribuzione Linux Ubuntu 10.04 LTS Server. Questa versione utilizza il sistema **Upstart** per la gestione dei task e servizi del sistema operativo. Il file di configurazione è **hsyco.conf**. Questo file si trova nella directory **/etc/init** e consente ad Upstart di gestire correttamente l'esecuzione di HSYCO.

La Chiave di Licenza Software

Il file **license.txt** contiene la chiave di licenza software necessaria per il funzionamento di HSYCO. Quando HSYCO viene avviato per la prima volta, o se questo file viene rimosso prima dell'avvio, HSYCO crea un nuovo file contenente un codice crittografato che identifica univocamente il computer sul quale il software viene eseguito. Per consentire la corretta esecuzione di HSYCO è necessario che questo file contenga anche la chiave di licenza software, rilasciata da Home Systems Consulting, specifica per ogni computer.

Il codice di identificazione hardware è calcolato automaticamente anche in base ai componenti hardware installati sul server. Ad esempio sostituendo o aggiungendo schede di rete il codice cambia e di conseguenza la chiave di licenza software non sarà più valida per la nuova configurazione.

Se la chiave di licenza non è presente nel file o non è corretta, HSYCO disabilita tutte le funzioni di accesso ai sistemi di campo, mantenendo in funzione solo il Web Server. Sarà quindi possibile accedere alle pagine configurate, ma non sarà possibile eseguire alcuna azione sugli impianti.

Gli eventuali messaggi di errore relativi alla licenza software vengono scritti nel file console.log.

Il contenuto di license.txt, generato da HSYCO e contenente solo il codice di identificazione hardware, è il seguente:

```
#Generated by HSYCO CONTROLLER 3.0.0
#Sat Dec 10 11:49:06 CET 2011
hwid=2080334b6283f450283bd4bbc3389bf66dc03dda
```

Per attivare le funzionalità complete di HSYCO sostituire il file license.txt generato da HSYCO alla prima attivazione, con la versione di license.txt completa della chiave fornita da Home Systems Consulting. La presenza del file

con la chiave viene riconosciuta automaticamente entro pochi secondi, non è quindi necessario riavviare HSYCO.

Un esempio del file completo della chiave di licenza software:

```
#Generated by HSYCO LICENSE MANAGER 3.0.0
#Sat Dec 10 12:20:00 CET 2011
name=Company Name
hwid=2080334b6283f450283bd4bbc3389bf66dc03dda
key=302D021500898B20F13ABE66DF3D2577809C6A53343A23A5CD0214540FD580
A1E3DA1CBEB18D11E98CAAE453D402BF
```


Accesso al Sistema Operativo

La configurazione del sistema operativo richiede un solo utente amministratore, la cui login è **hsyco**.

La password di default configurata per l'utente **hsyco** è **hsycoserver**.

Per garantire un'adeguata sicurezza e prevenire qualsiasi accesso non autorizzato è importante modificare la password di accesso di HSYCO prima di installare il sistema all'interno di una rete LAN non sicura.

L'accesso al sistema operativo avviene attraverso il protocollo sicuro SSH per l'accesso alla console di comando ed eventualmente per la gestione dei file. Il protocollo SSH protegge con la crittografia tutto il traffico e può essere utilizzato in modo sicuro anche per la gestione remota. In presenza di firewall sulla rete può essere necessario aprire la porta **TCP 22** e configurare il forward del traffico verso l'indirizzo IP di HSYCO SERVER.

HSYCO è installato nella home directory dell'utente **hsyco**, **/home/hsyco**.

In questa directory, una ulteriore sotto-directory **./hsyco** contiene tutto il software, i file di configurazione, programmazione, log e l'archivio delle immagini registrate.

In ambiente Microsoft Windows sono disponibili diversi software che consentono l'accesso SSH, in particolare **PuTTY**⁴ per l'accesso alla console e **Bitvise Tunnelier**⁵ e **WinSCP**⁶ per l'accesso SFTP ai file.

In ambiente Apple Mac OS X l'accesso SSH è integrato nel sistema operativo e utilizzabile direttamente attraverso Terminal, mentre per la gestione dei file mediante SFTP una delle utility più diffuse è **Cyberduck**⁷.

⁴ <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

⁵ <http://www.bitvise.com/download-area>

⁶ <http://winscp.net/eng/index.php>

⁷ <http://cyberduck.ch/>

Avviare e Fermare HSYCO

HSYCO si avvia automaticamente insieme al sistema operativo nel momento in cui il server viene alimentato. L'esecuzione di HSYCO è segnalata acusticamente da tre **beep** successivi a intervalli di un secondo.

HSYCO inoltre si riavvia automaticamente a seguito della modifica di uno dei file **hsyco.ini**, **hsyco.jar** e **com/hsyco/user.class**. Anche in questo caso il riavvio è segnalato da tre beep.

Per fermare manualmente l'esecuzione di HSYCO, accedere alla console via SSH:

```
ssh hsyco@192.168.0.50
```

ed eseguire il comando:

```
sudo stop hsyco
```

Per avviare HSYCO eseguire:

```
sudo start hsyco
```

Infine per verificare lo stato corrente di esecuzione di HSYCO eseguire

```
sudo status hsyco
```

Nel caso HSYCO sia in esecuzione la risposta al comando sarà simile a:

```
hsyco (start) running, process ...
```

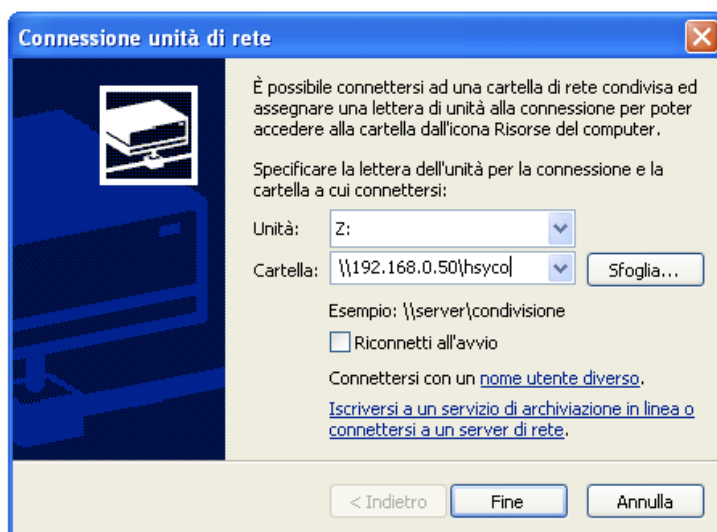

Accesso ai File

Per un più agevole accesso ai file, HSYCO SERVER rende disponibile la directory principale attraverso un servizio standard di *file server* (disco di rete). Il nome del servizio è *hsyco*. Per accedere al servizio si utilizza lo stesso utente *hsyco* e la password utilizzata per l'accesso al sistema operativo.

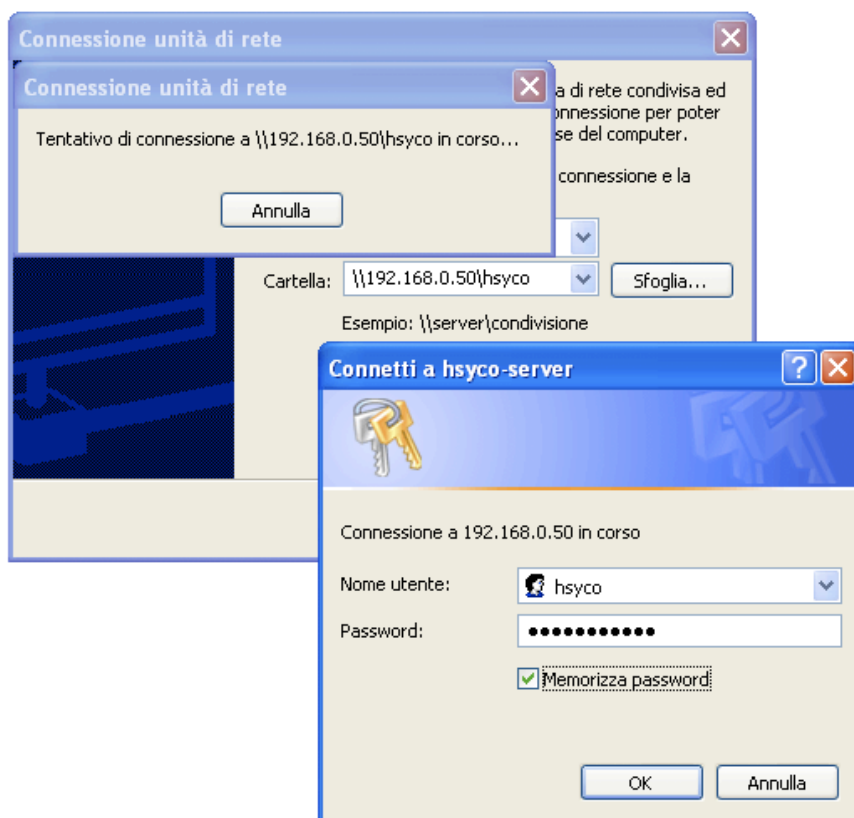
Accesso ai File da Microsoft Windows

Per connettere un sistema Microsoft Windows alla cartella di rete condivisa di HSYCO SERVER, selezionare **Strumenti > Connetti unità di rete** dal menu di **Esplora risorse**.

Nel campo **Cartella** inserire \\192.168.0.50\hsyco e premere il bottone **Fine**.



Inserire quindi il nome utente (sempre *hsyco*) e la password (inizialmente *hsycoserver*, modificabile come descritto in precedenza) e premere il bottone **OK**.



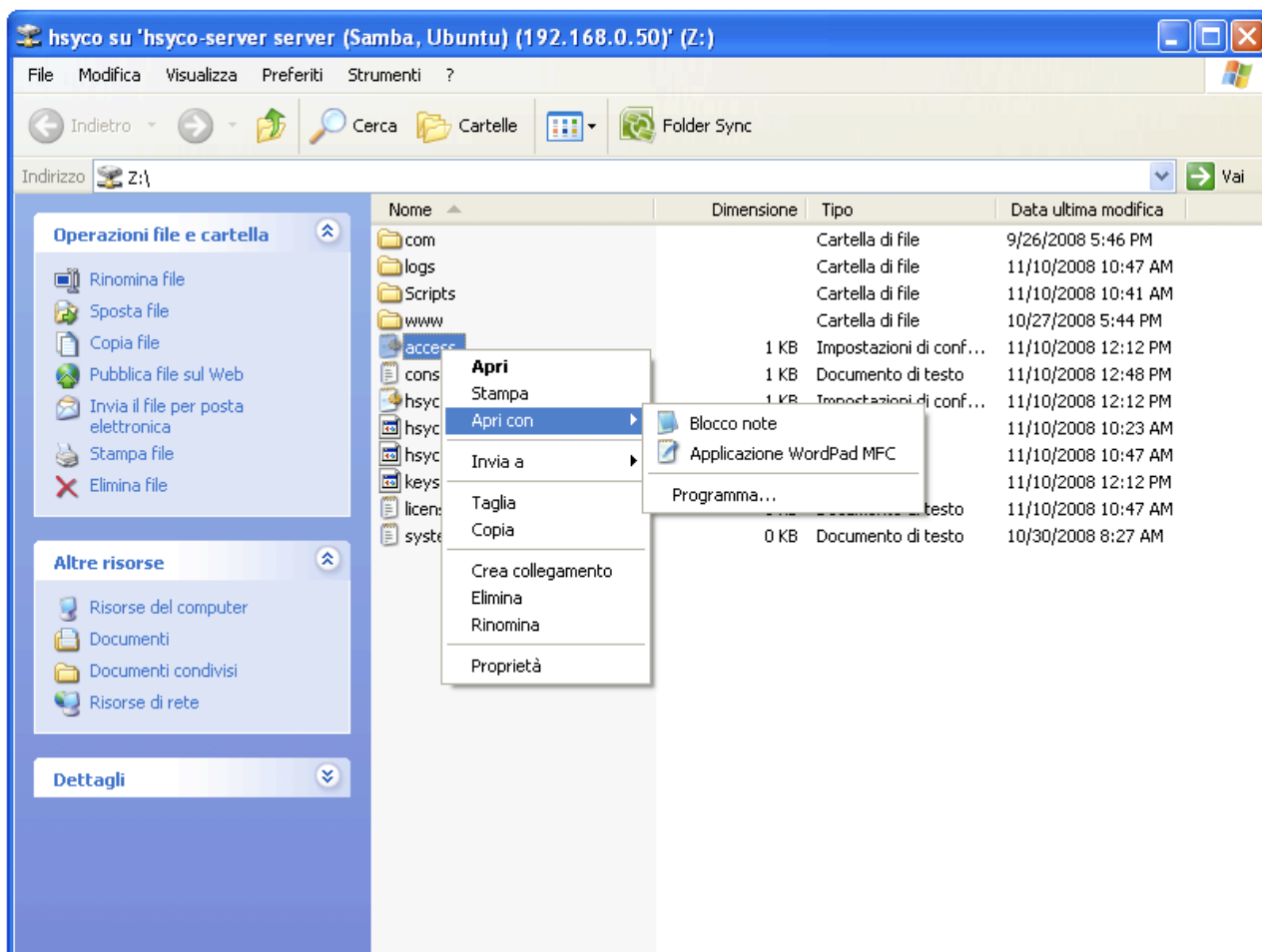
Nella finestra della cartella condivisa sono presenti e accessibili, sia in visualizzazione che in modifica, tutti i file di HSYCO.

Per garantire la massima sicurezza e protezione di HSYCO da accessi non autorizzati, si raccomanda di disconnettere la cartella condivisa quando non necessario e di non memorizzare⁸ la password per l'accesso automatico.

Per editare manualmente il contenuto dei file direttamente dalla cartella condivisa, utilizzare l'applicazione **WordPad**, non l'applicazione **Blocco note**, quest'ultima infatti non gestisce correttamente il formato dei file testo.

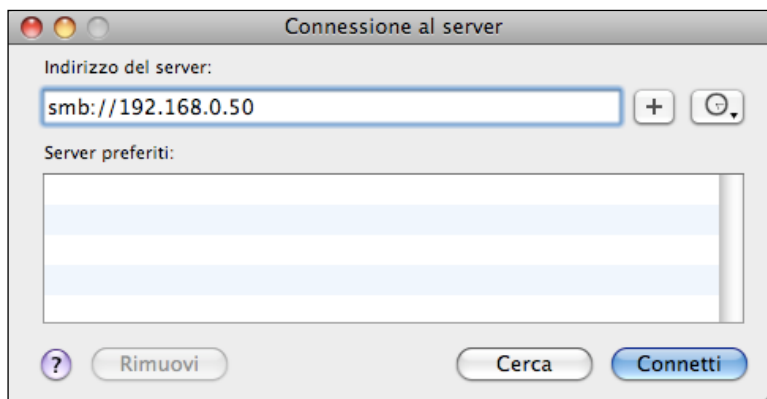
⁸ Non selezionare l'opzione *Memorizza password* nella finestra in cui vengono richiesti nome utente e password.

Per aprire i file utilizzando *WordPad*, cliccare sul file con il bottone destro, quindi selezionare **Apri con** ed infine *WordPad* se appare direttamente nel menu, oppure *Programma* e scegliere *WordPad* nella lista delle applicazioni disponibili.



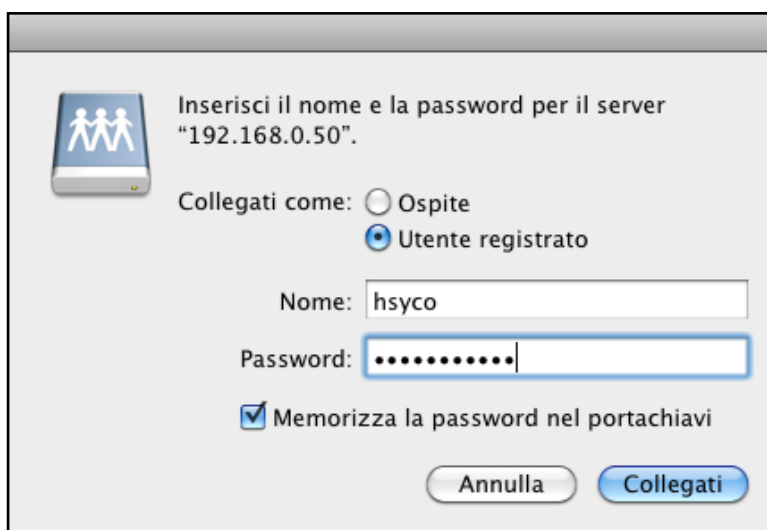
Accesso ai File da Mac OS X

Per connettere un sistema Apple Mac OS X al file server di HSYCO SERVER, selezionare **Vai > Connessione al Server** dal menu di **Finder**.



Nel campo **Indirizzo del Server** inserire `smb://192.168.0.50` e premere il bottone **Connetti**.

Inserire quindi il nome utente (sempre `hsyco`) e la password (inizialmente `hsycoserver`, modificabile come descritto in precedenza) e premere il bottone **Collegati**.



Per garantire la massima sicurezza e protezione di HSYCO da accessi non autorizzati, si raccomanda di disconnettere il file server condiviso quando non necessario e di non memorizzare la password nel *keychain* del Mac.

Configurazione

La configurazione di HSYCO è basata sul file ***hsyco.ini***.

Esso contiene i parametri di configurazione generale ed alcuni parametri specifici dei sistemi di campo gestiti.

Sicurezza e Gestione Utenti

La protezione di qualunque sistema da accessi non autorizzati è di fondamentale importanza. HSYCO utilizza diverse soluzioni di protezione, tali da renderlo estremamente sicuro sia nell'utilizzo attraverso la rete Internet ma anche all'interno della propria rete locale.

Nella configurazione standard HSYCO accetta solo accessi Web attraverso il protocollo sicuro **HTTPS**, mentre l'accesso utente via HTTP è disabilitato.

Inoltre, per contrastare sistemi automatizzati di service discovery, HSYCO non risponde alle richieste Web in cui sia definito solo l'indirizzo del server, come ad esempio `https://192.168.0.50`, ma richiede un URL più esteso, che comprenda una chiave di accesso, denominata **URLKey**, il cui valore è definito nel file `hsyco.ini`.

La URLKey non deve essere considerata come una password segreta, ma un elemento aggiuntivo di protezione.

E' necessario dunque modificare alcuni parametri del file `hsyco.ini` che interessano e condizionano proprio gli aspetti base della sicurezza e autenticazione degli utenti.

Impostazioni Iniziali di hsyco.ini

Il contenuto di hsyco.ini prima della configurazione è il seguente:

```
verboseLog = false
eventsLog = true
trustedNet = local
KeysTrustedValidityHours = 100000
KeysNotTrustedValidityHours = 1
AutoKillFiles = hsyco.ini,hsyco.jar,com/hsyco/user.class
openServersDiscovery = true
HTTPServerPort = 80
HTTPSSLServerPort = 443
ServerName = hsyco
URLKey = hsyncoserver
SysLogServerPort = 514
Latitude = 45
Longitude = 10
```

Senza entrare nel merito di tutti i parametri presenti, che saranno descritti in seguito, i parametri strettamente relativi alle impostazioni iniziali di sicurezza sono **ServerName** and **URLKey**.

In **trustedNet** è necessario specificare gli indirizzi IP considerati privati e sicuri. E' possibile inserire più gruppi di indirizzi separati da “,” o anche un solo indirizzo IP, o più semplicemente digitare “local” (in questo modo HSYCO considera sicuri tutti gli indirizzi IP della rete locale).

Gli indirizzi indicati vengono considerati relativamente sicuri da HSYCO. Per questi indirizzi può essere normalmente definito un time-out di sessione diverso, normalmente più lungo, rispetto a qualunque altro indirizzo Internet.

ServerName specifica il nome utilizzato per generare il certificato **SSL**, necessario per la crittografia del traffico Web HTTPS, e deve corrispondere al nome di dominio attraverso il quale il server HSYCO è accessibile via Internet.

Il certificato è contenuto nel file **hsyco.keys**. Quando HSYCO viene avviato, se questo file non è presente, viene generato automaticamente un nuovo certificato SSL in base al nome definito in **ServerName**. Altrimenti HSYCO si

limita a utilizzare il certificato contenuto in questo file, che può quindi anche essere stato generato da una **Certification Authority (CA)** ufficiale. A meno che non si sia scelto di utilizzare un certificato ufficiale, questo file viene quindi creato e gestito da HSYCO senza bisogno di interventi manuali.

Se il nome definito in ServerName viene modificato, la modifica ha effetto subito dopo il riavvio di HSYCO.

Prima di iniziare ad utilizzare HSYCO è necessario modificare la **URLKey**, sostituendo a hsyncoserver una nuova parola, composta da numeri e lettere maiuscole e minuscole di almeno 8 caratteri. E' possibile eventualmente specificare più URLKey, separate da virgola; in questo caso tutte le URLKey specificate saranno valide per l'accesso Web a HSYCO.

La URLKey deve essere lunga almeno 8 caratteri. Stringhe più corte verranno ignorate. In questo caso non sarà possibile accedere alle pagine Web di HSYCO.

Introduzione ai Sistemi di Campo

HSYCO si interfaccia con diversi sotto-sistemi di campo, che includono sistemi di illuminazione e automazione, videosorveglianza, sistemi di sicurezza, impianti di riscaldamento/condizionamento e altri specifici dispositivi. Inoltre, HSYCO supporta numerosi protocolli standard come BACnet, DALI, DMX, KNX, Modbus e altri protocolli proprietari.

Molti di questi sistemi di campo vengono rappresentati all'interno di HSYCO come "I/O Server". Un I/O Server è un modulo software che fornisce un livello di astrazione standardizzato in modo tale che anche sistemi diversi possano interfacciarsi utilizzando una nomenclatura convenzionale e un unico modello di programmazione.

Per maggiori dettagli sulla configurazione e per avere ulteriori informazioni sugli I/O Server consultare la documentazione contenuta nelle Application Notes.

La seguente tabella elenca tutti i sistemi attualmente supportati come I/O Server.

Tipo di Sistema	Descrizione
Airzone Innobus	sistema di controllo HVAC Airzone Innobus
Aritech	centrali antintrusione Aritech Comfort CSx75 e Master ATS
Aton	sistema audio multi-room Aton AH66T
Axis Decoder	controllo e streaming per l'AXIS Video Decoder
BACnet	client BACnet/IP
Dali	protocollo DALI attraverso il modulo interfaccia Tridonic SCI2
Daikin VRF	sistemi VRF Daikin con Controllore Intelligente
Duemmegi Contatto	sistema bus proprietario per sistemi di building automation
Duemmegi Domino	sistema bus proprietario per sistemi di home automation
Ekahau	server RTLS Ekahau
EL.MO.	centrale antintrusione EL.MO. ETR100
Elsner P03/3	versione Modbus della stazione meteo P03/3 Elsner

Tipo di Sistema	Descrizione
proiettori Epson ESC/VP	proiettori Epson di rete che supportano il protocollo di controllo ESC/VP
modem GSM	modem GSM standard per messaggistica SMS e eventi di chiamata vocale
Guardall	centrali antintrusione Guardall QX32i e PX80
HID Edge Solo	soluzione di rete EdgeReader per controllo accessi HID Edge Solo stand-alone
HSYCO remoto	accesso ai data point di HSYCO server remoti
HW Group - dispositivi I/O	I/O Controller e dispositivi ER-xx
HW Group - dispositivi di acquisizione dati	STE, Poseidon e Damocles di HW Group
Interlogix FP2000	centrale antincendio GE URC Interlogix basata sul protocollo FP2000
KNX/EIB	protocollo KNX/EIB attraverso il gateway Weinzierl KNX/IP BAOS 771/772 con Object Server 2.0
ISMG inverters	inverter ISMG Carlo Gavazzi
Mitsubishi M-NET	sistemi VRV Mitsubishi che supportano il protocollo proprietario M-NET
My Home	sistema bus di home automation My Home BTicino / Legrand
Modbus TCP servers	HSYCO agisce come client Modbus TCP e si connette a server configurati come gateway attraverso Modbus RTU RS485 o dispositivi slave Modbus ASCII
NuVo Concerto and Essentia	sistema audio multi-room NuVo'
PowerOne inverters	inverter Aurora PowerOne con supporto proprietario di comunicazione Aurora
RayCONTROL	sistemi di controllo HVAC di FCC Planterm
T-Lab	centrale antintrusione T-Lab Web 144
Yamaha RXV A/V receivers	sistemi audio/video di rete della famiglia RXV-x067 di Yamaha
Meteo Online	acquisizione Internet-based delle condizioni e previsioni meteo prelevate da Google

Un tipo di I/O Server speciale è il server DUMMY, che non si interfaccia a nessun sistema, ma semplicemente simula il comportamento di dispositivi di illuminazione e automazione e può essere utilizzato a scopi dimostrativi o per implementare una logica di controllo complessa, essendo un Server I/O virtuale.

I pochi sistemi integrati con HSYCO che non sono I/O Server, vengono supportati attraverso API dedicate, o perché necessitano di particolari requisiti o per problemi di compatibilità con le versioni precedenti di HSYCO. La seguente tabella elenca tutti i sistemi attualmente supportati attraverso una API specifica.

Tipo di Sistema	Descrizione
Bentel KYO 320	centrale di controllo antifurto
DMX	protocollo DMX attraverso il gateway KISS-BOX
telecamere IP	telecamere IP, con acquisizione JPEG e MJPEG di AXIS, Mobotix, Panasonic, Samsung, Vivotek, Rovio e qualsiasi altra telecamera IP che fornisce frame JPEG o flussi MJPEG via HTTP
IRTrans	emettitori/ricevitori di rete IR
Netstreams Musica	sistema audio multi-room Musica di Netstreams
Paradox EVO 48/192	centrale di controllo antifurto
RS-232	porte seriali locali RS-232 di HSYCO Server, e porte remote attraverso gateway seriali IP
Squeezebox	lettori musicali IP Squeezebox di Logitech
Tecnoalarm Tecno-Out	centrale di controllo antifurto
Wi-Fi Access Point	access point Wi-Fi di diversi produttori, utilizzati per fornire informazioni relative alla posizione dei client connessi alla Wi-Fi

Le impostazioni di configurazione per la maggior parte di questi sistemi sono da dichiarare all'interno del file di configurazione `hsyco.ini`. Gli altri sistemi richiedono invece un proprio specifico file di configurazione.

Il File hsyco.ini

E' il file principale di configurazione di HSYCO.

La modifica di qualunque parametro di questo file provoca dopo alcuni secondi il riavvio automatico di HSYCO, rendendo efficaci le modifiche effettuate.

Di seguito vengono elencati tutti i parametri presenti e descritte le modalità di configurazione delle diverse funzioni. Il formato del file è il seguente:

nome_parametro = valore

E' prevista la possibilità di inserire righe di commento, che vengono ignorate ai fini della configurazione. Una riga di commento è identificata dal carattere “#” a inizio riga. E' possibile separare le righe dei parametri con righe vuote per migliorare la leggibilità del file.

Controllo Accessi

Nome	Default	Formato
URLKey	hsycoserver	stringa di almeno 8 caratteri
Prima di iniziare ad utilizzare HSYCO è necessario modificare la URLKey, sostituendo a hsycoserver una nuova parola, composta da numeri e lettere maiuscole e minuscole di almeno 8 caratteri. E' possibile eventualmente specificare più URLKey, separate da virgola; in questo caso tutte le URLKey specificate saranno valide per l'accesso Web a HSYCO		

Nome	Default	Formato
WebAdminNetConfigLock	false	true false
Se true, le funzioni di gestione dei parametri di rete di HSYCO attraverso l'interfaccia Web sono disabilitate. Questo parametro dovrebbe essere impostato a true dopo la configurazione definitiva, se non si prevede siano necessarie ulteriori modifiche		

Nome	Default	Formato
trustedNet	local	nn.nn.nn.nn-nn.nn.nn.nn local

indirizzi IP che delimitano il gruppo di indirizzi di rete appartenenti alla rete locale sicura. E' possibile inserire più gruppi di indirizzi separati da “,” o anche un solo indirizzo IP, o semplicemente digitare “local”; in questo modo HSYCO assume che tutti gli indirizzi IP della rete locale siano sicuri.

I terminali che si collegano a HSYCO da questi indirizzi sono soggetti al time-out definito in KeysTrustedValidityHours, normalmente più lungo di quello utilizzato per tutti gli altri indirizzi IP, definito in KeysNotTrustedValidityHours

Nome	Default	Formato
KeysTrustedValidityHours	24	intero > 0 or hh:mm

time-out in ore per connessioni da indirizzi IP sicuri.

Può essere impostato ad un valore molto alto, es. 100000, per evitare di fatto che le sessioni scadano nel tempo. Può essere utilizzato anche il formato ore:minuti *hh:mm*

Nome	Default	Formato
KeysNotTrustedValidityHours	1	intero > 0 or hh:mm

time-out in ore per connessioni da indirizzi IP non sicuri.

Può essere utilizzato anche il formato ore:minuti *hh:mm*

Nome	Default	Formato
KeysInactivityHours		intero > 0 or hh:mm

time-out di inattività, in ore.

Questo parametro è opzionale. Se non impostato, il timer di inattività non sarà attivo e la sessione sarà disattivata solo in base al tempo trascorso dall'ultimo login.

Può essere utilizzato anche il formato ore:minuti *hh:mm*

Nome	Default	Formato
KeysInactivityMode	browser	browser cameras commands

modalità di funzionamento del timer di inattività.

Questo parametro è opzionale ed è significativo solo quando è definito anche il parametro *KeysInactivityHours*.

In modo *browser*, è sufficiente che il browser Web sia aperto sulla pagina di HSYCO per mantenere attiva la sessione.

In modo *cameras* la sessione rimane attiva solo eseguendo comandi o visualizzando le telecamere.

In modo *commands* la sessione rimane attiva solo eseguendo comandi entro il tempo massimo di inattività definito

Nome	Default	Formato
HTTPServerLowSecurityEnabled	false	true false

normalmente il server HTTP è attivo solo per consentire alle telecamere di inviare le notifiche di motion detection e ai sistemi PBX le notifiche delle chiamate.

Per evitare che le chiavi di autenticazione, i PIN e i PUK vengano trasmessi in chiaro, il protocollo HTTP non viene utilizzato per il normale accesso Web a HSYCO.

Impostare questo parametro a true solo se si vuole comunque abilitare il protocollo non sicuro HTTP per l'accesso Web a HSYCO

Nome	Default	Formato
HTTPServerPublicDirectory		nome della directory

se definito, abilita la funzione di Web Server pubblico per servire, senza autenticazione, i file presenti nella directory specificata. Il parametro imposta il nome della directory creata all'interno della cartella www, utilizzata per servire file pubblici. Se, per esempio, viene impostato il parametro HTTPServerPublicDirectory=public e un file dichiarato home.html all'interno della directory public, allora l'indirizzo https://192.168.0.50/public/home.html punterà a quel file (non è necessario aggiungere la URLKey).

Il Web Server pubblico risponde solo ai client che appartengono alla rete di indirizzi IP sicuri (trustedNet), via HTTP e HTTPS.

Livelli di Log

Nome	Default	Formato
eventsLog	false	true false
se true viene abilitato il log degli eventi rilevati dai sistemi di campo, ad esempio gli eventi legati agli I/O Server		

Nome	Default	Formato
userLog	false	true false
se true viene abilitato il log dei metodi Java richiamati in user.java o dei comandi definiti con la programmazione EVENTS		

Nome	Default	Formato
verboseLog	false	true false silent
se true viene abilitato il log esteso, utile per la diagnostica di anomalie o in fase di personalizzazione avanzata o sviluppo di codice Java.		
Se silent, vengono scritti su file solo gli errori gravi e disabilitati tutti gli altri messaggi, compresi gli eventi rilevati dal campo		

Alta Affidabilità

Nome	Default	Formato
haMode		master slave
<p>abilita la modalità di funzionamento in alta affidabilità di HSYCO.</p> <p>La configurazione ad alta affidabilità prevede l'utilizzo di due sistemi HSYCO distinti, uno deve essere configurato come sistema master, l'altro come slave</p>		

Nome	Default	Formato
haMasterIP	obbligatorio	indirizzo IP: nn.nn.nn.nn
<p>obbligatorio per la configurazione ad alta affidabilità.</p> <p>Specifica l'indirizzo IP di HSYCO in modalità master</p>		

Nome	Default	Formato
haSlaveIP	obbligatorio	indirizzo IP: nn.nn.nn.nn
<p>obbligatorio per la configurazione ad alta affidabilità.</p> <p>Specifica l'indirizzo IP di HSYCO in modalità slave</p>		

Nome	Default	Formato
haDisableCommandEvents	obbligatorio	true false
<p>obbligatorio per la configurazione ad alta affidabilità.</p> <p>Se true, il metodo <i>irtransCommand()</i> e l'azione IR in EVENTS e i comandi inviati agli I/O Server che supportano l'alta affidabilità vengono automaticamente disabilitati sull'HSYCO che non risulta attivo (ad esempio il sistema definito come slave non è attivo quando il sistema master funziona regolarmente).</p> <p>L'invio di comandi DMX dal sistema disattivo, generati da user.java o events.txt o dall'interfaccia Web è sempre disabilitato, anche quando questo parametro è impostato a false</p>		

Rete

Nome	Default	Formato
HTTPServerPort		numero intero < 65535
porta TCP/IP del Web Server HTTP di HSYCO. Se non presente, il server HTTP non viene abilitato. Normalmente impostato a 80		

Nome	Default	Formato
HTTPSSLServerPort		numero intero < 65535
porta TCP/IP del Web Server HTTPS di HSYCO. Se non presente, il server HTTPS non viene abilitato. Normalmente impostato a 443		

Nome	Default	Formato
OffLineCache	false	true false
HSYCO implementa la funzionalità di cache off-line dei browser Web che supportano lo standard HTML 5.0. Per abilitare la memorizzazione off-line sul browser Web delle pagine HTML di HSYCO, impostare questo parametro a true		

Nome	Default	Formato
ServerName	hsyco	nome di dominio
nome utilizzato per generare il certificato SSL, necessario per la crittografia del traffico Web HTTPS, e deve corrispondere al nome di dominio attraverso il quale il server HSYCO è accessibile via Internet. Il certificato si trova all'interno del file hsyco.keys. All'avvio di HSYCO, se questo file non è presente o non corrisponde al parametro ServerName impostato in hsyco.ini, un nuovo certificato SSL viene generato automaticamente in base al nome definito in ServerName		

Nome	Default	Formato
SysLogServerPort		numero intero < 65535
porta TCP/IP del Web Server SYSLOG di HSYCO. Il server SYSLOG è utilizzato per ricevere dagli Access Point (AP) le informazioni necessarie per determinare attraverso quale AP sia connesso ciascun utente presente nella rete LAN Wi-Fi. Normalmente è impostato a 514		

Orologio

Nome	Default	Formato
Latitude	0	numerico positivo negativo, con decimali separati da “.”
latitudine in gradi decimali (positivo per l'emisfero Nord), per il calcolo della posizione del Sole e degli orari di alba e tramonto		

Nome	Default	Formato
Longitude	0	numerico positivo negativo, con decimali separati da “.”
longitudine in gradi decimali (positivo per l'emisfero Est), per il calcolo della posizione del Sole e degli orari di alba e tramonto		

Nome	Default	Formato
SunriseOffsetMinutes	0	intero positivo negativo
deviazione in minuti dall'orario di alba astronomico (anticipo se minore di zero, ritardo se maggiore di zero)		

Nome	Default	Formato
SunsetOffsetMinutes	0	intero positivo negativo
deviazione in minuti dall'orario di tramonto astronomico (anticipo se minore di zero, ritardo se maggiore di zero)		

Nome	Default	Formato
TimeAutoUpdate	true	true false nome/indirizzo del server NTP
HSYCO imposta automaticamente ora/data locali in base ai dati provenienti dai server Network Time Protocol (NTP). Per disabilitare questa funzione, impostare a false il parametro. Per utilizzare uno specifico NTP server, impostare il parametro al nome o all'indirizzo IP del server NTP		

Nome	Default	Formato
TimeZone		ID del fuso orario

imposta il fuso orario corrente. Questo dato viene utilizzato per ottenere il fuso orario locale e regolare automaticamente l'ora in base all'ora legale. Se questo parametro non è impostato, HSYCO utilizzerà le impostazioni del fuso orario del sistema operativo.

L'id del fuso orario può essere dichiarato nel formato "Area/Location" o più genericamente "GMT+NN", "GMT-NN", formato che definisce l'offset delle ore rispetto all'ora in UTC.

Se viene utilizzato il formato "Area/Location", l'ora viene regolata automaticamente in base all'ora legale. Il valore TimeZone non deve contenere spazi; sostituire agli spazi il simbolo _, ad esempio "America/New_York".

E' possibile verificare le impostazioni del fuso orario corrente dal file message.log alla partenza di HSYCO, subito dopo il messaggio di start-ip, ad esempio:

```
2012.07.10 15:28:30.113 - HSYCO Ver. 3.1.0 Build 0097 (USER Ver. No User Code) started
2012.07.10 15:28:30.425 - Time zone is: Europe/Rome (Central European Summer Time)
```


DMX Gateway

Nome	Default	Formato
dmxServers		lista di nomi separati da virgola (solo lettere e numeri)
<p>lista degli identificativi associati ai gateway DMX512. Gli ID devono essere univoci e diversi dagli ID di altri I/O Server.</p> <p>Per ogni ID è necessario dichiarare anche il parametro <i>dmxServersIP.id</i></p>		

Nome	Default	Formato
dmxServersIP.id	obbligatorio per KissBox	indirizzo IP: nn.nn.nn.nn
<p>indirizzo IP del gateway DMX KissBox. Sostituire <i>id</i> con uno dei nomi dichiarati nel parametro <i>dmxServers</i></p>		

Nome	Default	Formato
dmxServersPort.id	obbligatorio per KissBox	numero intero < 65535
<p>porta TCP/IP del gateway DMX KissBox</p>		

Nome	Default	Formato
dmxServersComm.id	obbligatorio per ENTTEC	stringa
<p>nome della porta seriale virtuale associata al controllore ENTTEC DMX USB PRO. Consultare l'Application Note dell'ENTTEC DMX USB PRO per ulteriori informazioni</p>		

Telecamere

Nome	Default	Formato
Cameras		lista di nomi separati da virgola

questo parametro è necessario per l'acquisizione di immagini da telecamere IP. Contiene la lista dei nomi identificativi delle telecamere. Gli ID devono essere univoci e diversi dagli ID di altri I/O Server.

Per ogni identificativo definito deve essere presente un ulteriore parametro, *Camera.id.URL*, e altri parametri opzionali

Nome	Default	Formato
CamerasRecordingMotionTriggerSeconds	5	numero intero positivo

numero di secondi durante i quali HSYCO continua a registrare da una telecamera in seguito ad un evento di motion detection ricevuto dalla telecamera stessa. E' possibile impostare il valore di questo parametro a 0, disattivando la registrazione automatica degli eventi; questa soluzione può essere utile nel caso in cui si preferisca gestire la registrazione degli eventi direttamente da codice utente Java o da EVENTS

Nome	Default	Formato
CamerasRefreshMillis	1000	numero intero positivo

intervallo di acquisizione dei frame dalle telecamere, in millisecondi.

Ad esempio, per una frequenza di 4 frame al secondo, impostare il valore a 250. Ad un valore basso di questo parametro corrisponde un numero di frame più alto acquisiti ed eventualmente registrati nell'unità di tempo, con un maggior peso in termini di risorse di CPU e spazio occupato su disco

Nome	Default	Formato
CamerasResizedQuality	0.7	numero decimale compreso tra 0 e 1 (con decimali separati da .)

qualità dell'algoritmo di elaborazione delle immagini, utilizzato per la trasformazione delle immagini a risoluzione ridotta. Un numero prossimo a 1 produce una migliore qualità ma peso e dimensioni maggiori per ciascun frame

Nome	Default	Formato
Camera.id.URL	obbligatorio per ciascuna telecamera	URL completo

URL completo, dal quale è possibile acquisire i singoli frame in formato JPEG alla risoluzione desiderata.

La registrazione si basa sui frame acquisiti con questo URL.

Quando si utilizza il flusso MJPEG, è consigliabile configurare il numero di frame al secondo al valore più vicino alla frequenza di acquisizione dei frame di HSYCO come impostato nel parametro CameraRefreshMillis

Nome	Default	Formato
Camera.id.URL.Small		URL completo

ai fini di ottimizzare le performance dell'elaborazione delle immagini, è possibile impostare questo parametro opzionale, definendo un URL che cattura i frame a una risoluzione ridotta. Nell'invio di frame all'interfaccia Web, HSYCO sceglierà automaticamente la risoluzione dei frame più appropriata, basata sulla dimensione del frame nell'interfaccia Web. I frame più piccoli sono di solito usati per creare griglie, migliorando le prestazioni di elaborazione

Nome	Default	Formato
Camera.id.PTZ	0	nome del driver PTZ

definisce la modalità PTZ di una telecamere e la famiglia di driver PTZ. Il controllo della telecamera viene automaticamente aggiunto all'interfaccia Web. E' possibile cliccare ai lati dell'immagine per controllarne i movimenti: alto, basso, destra, sinistra, zoom e messa a fuoco. I tipi supportati sono i seguenti:

Camera.id.PTZ = axis (telecamere AXIS PTZ)

Camera.id.PTZ = axis-vptz (telecamere fisse con supporto PTZ virtuale AXIS)

Camera.id.PTZ = panasonic (telecamere serie BB e BL Panasonic)

Camera.id.PTZ = panasonic-wv (telecamere serie WV Panasonic)

Camera.id.PTZ = mobotix (telecamere Mobotix)

Camera.id.PTZ = rovio (telecamera WowWee Rovio)

Camera.id.PTZ = SNV-3120, SNP-3120, SNP-3120V, SNP-3120H (telecamere Samsung)

Camera.id.PTZ = vptz (telecam. fisse con supporto PTZ virtuale implementato da HSYCO)

Camera.id.PTZ = user (per associare codice Java alle zone sensibili dell'immagine)

Name	Default	Format
Camera.id.Type		stringa (lettere e numeri)

Per le telecamere che supportano questa opzione è possibile definire il tipo di modello di telecamera. Ad esempio, *Camera.samsung.Type = SND-5080*

Name	Default	Format
Camera.id.IO		enabled

E' possibile abilitare le opzioni I/O e VA (Visual Analysis) per una telecamera. Per esempio, *Camera.<camid>.IO=enabled*. Per abilitare la sola opzione di I/O, scrivere "*enabled:io*", oppure "*enabled:va*" per abilitare la sola opzione di VA

Nome	Default	Formato
Camera.id.User		stringa (lettere e numeri)

quando questo parametro è definito, le richieste di accesso ai frame e i comandi per il controllo PTZ della telecamera vengono inviati con autenticazione basata su user e password

Nome	Default	Formato
Camera.id.Password		stringa (lettere e numeri)

quando questo parametro è definito, le richieste di accesso ai frame e i comandi per il controllo PTZ della telecamera vengono inviati con autenticazione basata su user e password

Nome	Default	Formato
Camera.id.DroppedFrames	0	numero intero >= 0

questo parametro specifica il numero di frame da scartare durante le registrazioni.

Ad esempio, impostando il valore 1, verranno registrati metà dei frame rispetto a quelli normalmente acquisiti dalle telecamere; impostando il valore 3 verranno scartati 3 frame per ogni frame acquisito, registrando quindi solo 1/4 dei frame acquisiti. Scartare le frame riduce notevolmente lo spazio su disco e consente di ottenere una riproduzione più veloce delle registrazioni video

Nome	Default	Formato
Camera.id.MaxAge	30d	numero intero positivo seguito dal carattere "d" per giorni, <i>h</i> per ore oppure <i>m</i> per minuti

periodo di tempo, a partire dal momento della registrazione di ciascun frame acquisito dalla telecamera corrispondente a id, durante il quale i frame rimangono accessibili.

Superato questo tempo i frame vengono automaticamente cancellati dalla directory *motion* di HSYCO.

I frame più vecchi possono essere cancellati automaticamente anche prima del tempo limite di conservazione, nel caso in cui lo spazio disponibile su disco scenda al di sotto di un valore minimo di sicurezza

Nome	Default	Formato
Camera.id.MotionBuffer	0	numero intero ≥ 0

in corrispondenza di un comando di registrazione delle immagini (ad esempio in seguito a motion detection) HSYCO può registrare un numero di frame precedenti all'evento, definiti da questo parametro.

Quando impostato, HSYCO acquisisce costantemente le immagini dalla telecamera, non solo quando necessario. Inoltre la registrazione temporanea di questo buffer richiede risorse di memoria, è quindi conveniente non utilizzare valori elevati (valori compresi tra 10 e 20, con intervallo di acquisizione di circa 400 millisecondi, possono essere considerati normali). Impostare il buffer solo quando effettivamente utile

Nome	Default	Formato
Camera.id.RemoteRequestPassword		stringa (lettere e numeri)

HSYCO è in grado di servire le immagini delle telecamere attraverso una richiesta HTTP protetta da password, ad esempio:

`https:<hsycoserver>/x/camera/<cameraid>?password=<password>&size=<width>x<height>`
(il parametro size è opzionale) per frame singoli oppure:

`https://<hsycoserver>/x/camerastream/<cameraid>?size=<width>x<height>&password=<pwd>[&period=<millis>]` per stream MJPEG.

E' possibile specificare password differenti per richieste provenienti dall'interno del gruppo di indirizzi IP sicuri e dall'esterno

Nome	Default	Formato
Camera.id.TrustedRequestPassword		stringa (lettere e numeri)
Simile a RemoteRequestPassword, è utilizzato per definire password di accesso distinte per le richieste provenienti da indirizzi IP appartenenti al gruppo degli indirizzi sicuri, definiti con il parametro trustedNet. Le richieste di questo tipo vengono servite anche attraverso il protocollo HTTP, senza protezione SSL		

Nome	Default	Formato
Camera.id.Rotate	0	numero intero
questo parametro consente di ruotare le immagini ricevute dalle telecamere. Corrisponde al numero di gradi decimali di rotazione. Un numero positivo produce una rotazione in senso orario. A seguito della rotazione le immagini potranno risultare tagliate ed eventualmente presentare delle barre ai bordi per adattarsi alle dimensioni dell'immagine visualizzata. L'impostazione di questo parametro potrebbe causare un notevole degrado delle prestazioni, in particolar modo durante l'elaborazione di immagini ad alta risoluzione		

Nome	Default	Formato
CameraGrid.id		lista di id di telecamere
<p>con questo parametro è possibile specificare fino a 99 diverse combinazioni di telecamere per realizzare pagine di visualizzazione a matrice. id deve essere compreso tra 1 e 99; le diverse combinazioni devono essere identificate progressivamente a partire da 1.</p> <p>Per ogni combinazione si specifica la lista delle telecamere riga per riga da sinistra a destra e dall'alto in basso rispetto alla matrice di visualizzazione, separando gli id delle telecamere della stessa riga con spazi e le righe tra loro con virgola.</p> <p>Ad esempio, per realizzare una matrice 2X2:</p> <p style="text-align: center;"><i>CameraGrid.1 = c1 c2, c3 c4</i></p> <p>e per una matrice 3X3:</p> <p style="text-align: center;"><i>CameraGrid.2 = c1 c2 c3, c4 c5 empty, c6 c7 c8</i></p> <p>La parola <i>empty</i> può essere utilizzata al posto di un id per indicare una posizione vuota</p>		

Nome	Default	Formato
CameraGrid.<i>id</i>.Resolution		larghezza X altezza in pixel

questo parametro opzionale viene usato per ottimizzare le performance di elaborazione delle griglie, in particolare di griglie complesse nel caso di skin di grandi dimensioni per l'interfaccia Web.

E' possibile definire una risoluzione massima per griglia. HSYCO mostrerà sempre questa griglia a una risoluzione che non è mai maggiore della larghezza e altezza definita in questo parametro.

Lo scopo è quello di ridurre la grandezza in byte dell'immagine, migliorando le prestazioni su connessioni remote, e di ridurre il carico di elaborazione sulla CPU richiesto per generare la griglia, con un potenziale beneficio sulla velocità dei frame.

Per esempio, per impostare una risoluzione massima di 640x480 pixel per la griglia 3:

CameraGrid.3.Resolution = 640x480

Nome	Default	Formato
CameraGrid.<i>N</i>.RemoteRequestPassword		stringa (lettere e numeri)

HSYCO è in grado di servire le immagini live delle griglie delle telecamere attraverso una richiesta HTTP protetta tramite password, ad esempio:

`https:<hsycoserver>/x/camera/<cameraid>?password=<password>&size=<width>x<height>`
(il parametro size è opzionale) per frame singoli oppure:

`https://<hsycoserver>/x/camerastream/<cameraid>?size=<width>x<height>&password=<pwd>[&period=<millis>]` per stream MJPEG.

E' possibile specificare password differenti per richieste provenienti dall'interno del gruppo di indirizzi IP sicuri e dall'esterno

Nome	Default	Formato
CameraGrid.<i>N</i>.TrustedRequestPassword		stringa (lettere e numeri)

Simile a RemoteRequestPassword, è utilizzato per definire password di accesso distinte per le richieste provenienti da indirizzi IP appartenenti al gruppo degli indirizzi sicuri, definiti con il parametro trustedNet. Le richieste di questo tipo vengono servite anche attraverso il protocollo HTTP, senza protezione SSL

I/O Server

Nome	Default	Formato
ioServers		lista di nomi separati da virgola (solo lettere e numeri)
lista degli identificativi associati agli I/O Server. Gli ID possono essere assegnati a discrezione. Per ogni identificativo definito devono essere specificati anche i parametri <i>ioServersIP.id</i> e <i>ioServersType.id</i> descritti di seguito		

Nome	Default	Formato
ioServersIP.id	obbligatorio	indirizzo IP: nn.nn.nn.nn
indirizzo IP di una interfaccia I/O. Sostituire <i>id</i> con uno dei nomi definiti nel parametro <i>ioServers</i> . Obbligatorio per ogni nome definito in <i>ioServers</i>		

Nome	Default	Formato
ioServersPort.id		numero intero < 65535
porta TCP/IP dell'I/O Server specificato. Utilizzare questo parametro per impostare una porta specifica per ogni I/O Server		

Name	Default	Formato
ioServersComm.id		stringa (lettere e numeri)
id della porta di comunicazione usata dall'I/O Server. Da utilizzare solo nel caso in cui il parametro <i>ioServer IP</i> non sia definito		

Name	Default	Format
ioServersType.id	obbligatorio	identificativo del tipo di interfaccia
tipo di I/O Server. Consultare le Application Notes per i tipi supportati. Obbligatorio per ogni I/O Server dichiarato		

Nome	Default	Formato
ioServersOptions.id		stringa (lettere e numeri)

Nome	Default	Formato
alcuni I/O Server richiedono parametri di configurazione aggiuntivi, come descritto nelle Application Notes		

Nome	Default	Formato
ioServersAuth.id.User		stringa (lettere e numeri)
alcuni I/O Server richiedono parametri di configurazione aggiuntivi, come descritto nelle Application Notes		

Nome	Default	Formato
ioServersAuth.id.Password		stringa (lettere e numeri)
alcuni I/O Server richiedono parametri di configurazione aggiuntivi, come descritto nelle Application Notes		

Server HSYCO Remoti

Nome	Default	Formato
RemoteServerPassword		stringa (lettere e numeri)
impostare questa password per abilitare l'accesso a un HSYCO server da parte di un altro I/O Server HSYCO remoto (per maggiori dettagli consultare l'Application Note relativa)		

Nome	Default	Formato
RemoteServerAddress		lista di indirizzi IP separati da virgola
imposta gli indirizzi IP dei server HSYCO remoti abilitati all'accesso di HSYCO server		

Nome	Default	Formato
RemoteServerControl	false	true o false
impostato a true consente al server remoto di modificare i valori dei data point di HSYCO server		

Porte di Comunicazione Seriale

Nome	Default	Formato
CommPorts		lista di nomi separati da virgola (solo lettere e numeri)

questo parametro contiene la lista dei nomi identificativi delle porte seriali gestite da HSYCO. Gli identificativi possono essere assegnati a discrezione.

Per ogni identificativo definito devono essere specificati i parametri *CommPort.id.Id*, *CommPort.id.Type* e *CommPort.id.Params*, dove *id* è il nome identificativo della porta.

Se *verboseLog* = true e *userLog* = true, i byte ricevuti e trasmessi sulla porta vengono trascritti nel file di log

Nome	Default	Formato
CommPort.id.Id	obbligatorio	nome di sistema della porta di comunicazione

deve essere impostato con il nome di sistema completo o parziale della porta di comunicazione associata. Ad esempio *ttyS0* oppure */dev/ttyS0*. Obbligatorio per ogni nome definito in *CommPorts*.

Per le porte seriali di I/O server di rete, impostare questo parametro con l'id che identifica il dispositivo di I/O (specificato in *ioServers*).

Per le porte seriali su *IRTrans*, impostare questo parametro con l'id che identifica il dispositivo (specificato in *IRTrans*)

Nome	Default	Formato
CommPort.id.Type	obbligatorio	"serial" "server" "io" "irtrans"

definisce il nome del driver utilizzato, corrispondente al tipo di porta associata. Per porte seriali locali, impostare al valore "serial". Obbligatorio per ogni nome definito in *CommPorts*.

Per le porte seriali remote che usano un convertitore IP seriale supportato, impostare il valore a "server", e i parametri *CommPort.id.IP* e *CommPort.id.Port* con l'IP del convertitore e la porta.

Per le porte seriali basate su server di rete I/O, impostare al valore "io" ed impostare *CommPort.id.Id* con lo stesso id che identifica il dispositivo di I/O.

Per le porte seriali su *IRTrans*, impostare il valore a "irtrans" e impostare il parametro *CommPort.id.Id* con l'id dell'*IRTrans*

Nome	Default	Formato
CommPort.id.Params	obbligatorio per ogni nome definito in CommPorts	parametri di comunicazione, separati da virgola

definisce i parametri di comunicazione per ciascuna porta seriale.

Per porte seriali locali i parametri sono obbligatoriamente: velocità, bit, stop, parità, controllo di flusso. I parametri devono essere impostati con valori supportati dai driver e dall'hardware di ciascuna porta.

La velocità è espressa come numero intero (baudrate). Il numero di bit dati può essere 8, 7, 6 o 5. Il numero di stop bit può essere 1, 1.5 o 2. La parità può essere 0 (nessuna parità), 1 (parità dispari), 2 (parità pari), 3 (parità mark) oppure 4 (parità space). Il controllo di flusso può essere 0 (nessun controllo), 1 (XON/XOFF) o 2 (RTS/CTS).

Questo parametro non è significativo per le porte seriali su IRTrans

Nome	Default	Formato
CommPort.id.IP	obbligatorio per ogni comm port ID definito di tipo "server"	nnn.nnn.nnn.nnn oppure, per configurazioni in failover: nnn.nnn.nnn.nnn, mmm.mmm.mmm.mmm

indirizzo IP del gateway IP seriale. Sostituire *id* con uno degli ID dichiarati in *CommPorts*

Nome	Default	Formato
CommPort.id.Port	obbligatorio per ogni comm port ID definito di tipo "server"	intero positivo < 65535

porta del converter IP seriale. Sostituire *id* con uno degli ID dichiarati in *CommPorts*

Nome	Default	Formato
CommPortsList	optional	lista dei nomi di sistema delle porte seriali locali separati da virgola

Nome	Default	Formato
<p>questo è un parametro opzionale, e non è richiesto a meno che il sistema operativo fallisca a rilevare automaticamente la porte seriali locali. In questo caso, è necessario elencare tutte le porte seriali locali utilizzando i nomi delle porte del sistema operativo, ad esempio: CommPortsList=/dev/ttyS0,/dev/ttyS1</p>		

IRTrans

Nome	Default	Formato
IRTrans		lista di nomi separati da virgola (solo lettere e numeri)
<p>questo parametro contiene la lista dei nomi identificativi dei moduli IRTrans presenti in rete e gestiti da HSYCO per l'invio o la ricezione di comandi IR (infrarossi).</p> <p>Gli identificativi possono essere assegnati a discrezione.</p> <p>Per ogni identificativo definito deve essere presente l'ulteriore parametro <i>IRTransIP.id</i></p>		

Nome	Default	Formato
IRTransIP.id	obbligatorio per ciascun modulo IRTrans definito	indirizzo IP: nn.nn.nn.nn
<p>indirizzo IP del modulo IRTrans corrispondente al nome assegnato.</p> <p>Sostituire <i>id</i> con uno dei nomi di IRTrans definiti nel parametro <i>IRTrans</i></p>		

Squeezebox

Nome	Default	Formato
slimServerName		indirizzo IP: nn.nn.nn.nn
indirizzo IP del server SqueezeCenter per il controllo degli Squeezebox presenti in rete		

Nome	Default	Formato
slimPlayers		lista di nomi separati da virgola (solo lettere e numeri)
<p>questo parametro contiene la lista dei nomi identificativi degli Squeezebox presenti in rete e gestiti da HSYCO. Gli identificativi possono essere assegnati a discrezione.</p> <p>Per ogni identificativo definito deve essere presente l'ulteriore parametro <i>slimPlayerId.id</i></p>		

Nome	Default	Formato
slimPlayerId.id	obbligatorio per ogni Squeezebox	stringa di identificazione del player fornita dal server SqueezeCenter
<p>stringa di identificazione dello Squeezebox corrispondente al nome assegnato, definita dal server SqueezeCenter. Normalmente si tratta del MAC Address di ciascun player, nel formato <i>hh:hh:hh:hh:hh:hh</i>, dove hh sono coppie di numeri 0-9 e/o lettere a-f. Sostituire id con uno dei nomi di Squeezebox definiti nel parametro <i>SlimPlayers</i></p>		

Public Announcement

Nome	Default	Formato
AudioServerTTS		stringa
questo parametro è obbligatorio solo nel caso in cui si utilizzi un motore text-to-speech diverso da quello di default del sistema operativo. Per maggiori dettagli consultare l'Appendice Public Announcement		

Nome	Default	Formato
AudioServerVolume		numero intero
cambia il volume in uscita impostato di default dal motore text-to-speech dei file audio		

Nome	Default	Formato
AudioServerQuality		numero intero
cambia la qualità audio di default impostata dal motore text-to-speech		

Nome	Default	Formato
AudioServerSpeed		numero intero
cambia la velocità di default impostata dal motore text-to-speech		

Timer

Nome	Default	Formato
Timers		lista di nomi separati da virgola (solo lettere e numeri)
questo parametro contiene la lista dei nomi identificativi dei timer definiti dall'utente. Un timer può essere impostato attraverso l'interfaccia Web, e genererà eventi di attivazione/disattivazione del timer sulla base della programmazione. Gli identificativi possono essere assegnati a discrezione.		

Servizi di Localizzazione

Nome	Default	Formato
LocationBases		lista di nomi separati da virgola (solo lettere e numeri)
<p>questo parametro, necessario per il funzionamento dei servizi di localizzazione degli utenti presenti nella rete LAN Wi-Fi, deve contenere la lista dei nomi identificativi degli Access Point presenti in rete.</p> <p>Gli identificativi possono essere assegnati a discrezione. Per ogni identificativo definito deve essere presente l'ulteriore parametro <i>LocationBaseIP.id</i> descritto di seguito, dove <i>id</i> è il nome identificativo. Questi nomi sono anche utilizzati nel file systemtopo.txt per associare a ciascuna location una descrizione definita dall'utente che comparirà nell'interfaccia Web</p>		

Nome	Default	Formato
LocationBaseIP.id	obbligatorio	indirizzo IP: nn.nn.nn.nn
<p>indirizzo IP dell'Access Point corrispondente al nome assegnato. Obbligatorio per ogni nome definito in LocationBases</p>		

Data Loggers

Nome	Default	Formato
DataLoggers		lista di ID separati da virgola (solo lettere e numeri)
lista degli ID per ogni data logger. Gli identificativi possono essere assegnati a discrezione. Per ogni id è necessario dichiarare anche il parametro <code>DataLoggers.id.Type</code>		

Nome	Default	Formato
DataLoggerCsvSeparator	comma	tab comma semicolon
definisce il tipo di separatore di campo per file CSV generati dai data loggers		

Nelle tabelle seguenti sostituire *id* con uno dei nomi dichiarati in DataLoggers.

Nome	Default	Formato
DataLoggers.id.Type	obbligatorio	counter range
tipo di datalogger, utilizzare “counter” o “range”		

Name	Default	Format
DataLoggers.id.UseCharts	false	true false
impostato a true il data logger imposta automaticamente gli oggetti chart e i totali. Impostato a false il data logger è utilizzato solo per raccolta dati e creazione dei file storici, o per la visualizzazione dei dati attraverso l'oggetto (datalogger)		

Nome	Default	Formato
DataLoggers.id.Decimals	1	numero intero positivo
numero di cifre decimali da usare per i dati raccolti		

Nome	Default	Formato
DataLoggers.id.HourInterval	1	numero intero positivo
opzione per i grafici orari. Specifica il numero di ore da raggruppare in un intervallo singolo. Per esempio, impostando un valore pari a 6, i grafici verranno divisi in gruppi di 6 ore: 0-5, 6-11, 12-17 e 18-23.		

Nome	Default	Formato
DataLoggers.id.CounterUpperLimit	0	numero float

opzione per data logger di tipo counter. Specifica il valore massimo raggiungibile dal valore di input (contatore). Un valore pari a 0 indica che nessun limite è impostato

Name	Default	Format
DataLoggers.id.CounterMaxDelta		numero float

opzione per data logger di tipo counter. Specifica il delta massimo ammissibile tra due letture consecutive. Se il delta calcolato supera tale valore, il valore letto viene ignorato. Se non specificato, tutti i valori sono accettati.

Si consiglia vivamente di impostare un limite delta al fine di evitare serie di dati che potrebbero risultare corrotte in caso di false letture

Nome	Default	Formato
DataLoggers.id.SeparateCharts	false	true false

opzione per data logger di tipo counter che usano fasce orarie. Specifica se utilizzare o meno grafici mensili e annuali separati per ogni fascia oraria definita

Nome	Default	Formato
DataLoggers.id.SlotAlign	true	true false

opzione per data logger di tipo counter che usano fasce orarie con grafici separati. Specifica se allineare o meno la scala dei grafici delle fasce appartenenti allo stesso periodo

Nome	Default	Formato
DataLoggers.id.RatesLog File		file path. e.g.: hsyco/rates.csv

opzione per data logger di tipo counter che usano fasce orarie. Specifica il percorso del file dove salvare i dati elaborati. Se l'opzione è omessa, non verrà creato alcun file di log.

Se il percorso specificato include le stringhe "%Y", "%M" o "%D", esse verranno sostituite rispettivamente dall'anno, mese o giorno corrente

Nome	Default	Formato
DataLoggers.id.PeriodAlign	true	true false
opzione per data logger di tipo range. Specifica se allineare o meno la scala dei grafici (min, max e avg) appartenenti allo stesso periodo		

Nome	Default	Formato
DataLoggers.id.Origin	0	float number
opzione per data logger di tipo range. Specifica il valore dell'origine dei grafici.		

Nome	Default	Formato
DataLoggers.id.Range		range: min:max
<p>opzione per data logger di tipo range. Specifica il range dei grafici come “min:max”, dove “min” indica il limite inferiore e “max” quello superiore. I limiti possono essere definiti come rigidi o flessibili: se il valore di un limite è seguito da “!” (es. “0!:20!”) allora è rigido, cioè non viene modificato anche se il data logger ha elaborato valori che eccedono tale limite. Al contrario, limiti flessibili (es. 0:20) vengono modificati ogni volta che un valore eccede tali limiti.</p> <p>Se il parametro è omissso, il range dei grafici spazierà dal valore minimo processato al valore massimo.</p>		

Nome	Default	Formato
DataLoggers.id.OutOfRangeMode	cut	cut ignore
<p>opzione per data logger di tipo range con limiti rigidi. Specifica il comportamento del data logger nel caso elabori valori che eccedono il range specificato: impostando il valore a “cut” tali valori verranno adattati ai limiti, altrimenti, impostando il valore a “ignore”, tali valori verranno ignorati. In entrambi i casi un messaggio di errore viene riportato per valori che eccedono i limiti.</p>		

Email

Nome	Default	Formato
SmtName		nome o indirizzo IP del server SMTP
<p>nome del server SMTP o indirizzo numerico.</p> <p>Questo parametro viene utilizzato dall'azione MAIL e dal metodo Java sendMail() per inviare email attraverso un account SMTP specifico e sicuro piuttosto che direttamente attraverso il server SMTP di destinazione</p>		

Nome	Default	Formato
SmtPort	25, 465 or 587	numero intero positivo
<p>porta del server SMTP, se diversa dalla porta di default</p> <p>La porta di default è:</p> <ul style="list-style-type: none"> • 25 for not encrypted traffic • 465 for SMTP over SSL • 587 for ESMTP 		

Nome	Default	Formato
SmtUser		stringa
username dell'account necessario per autenticare la connessione al server SMTP		

Nome	Default	Formato
SmtPassword		stringa
password dell'account necessaria per autenticare la connessione al server SMTP		

Nome	Default	Formato
SmtSSL	false	false true esmtpl
<p>“true” per il server SMTP che supporta SMTP over SSL (465 è la porta di default).</p> <p>“esmtpl” per il server SMTP che supporta ESMTP (587 è la porta di default).</p> <p>“false” per i server SMTP che non supportano il traffico criptato</p>		

Altri Parametri

Nome	Default	Formato
AutoKillFiles		lista di nomi di file separati da virgola
questo parametro viene solitamente impostato come segue: <i>hsyco.ini,hsyco.jar,com/hsyco/user.class</i> forzando così il riavvio automatico in seguito alla modifica dei tre file elencati		

Nome	Default	Formato
Language	en	en fr it
Alcuni I/O Server ed altri servizi di base generano automaticamente messaggi di testo. Questo parametro definisce la lingua dei messaggi generati da questi servizi		

Personalizzazione

La personalizzazione di HSYCO consiste di base nell'adattamento del sistema all'architettura dell'impianto gestito e in particolare nella personalizzazione dell'interfaccia Web per il controllo e comando dei sistemi.

HSYCO prevede un formato grafico standard, detto anche skin, ottimizzato per garantire un funzionamento omogeneo ed efficiente su un numero elevato di dispositivi, dai vari Web Browser per PC a molti dispositivi portatili.

Il modo più semplice ed immediato di personalizzare HSYCO, che non richiede alcuna competenza di programmazione o di sviluppo di pagine HTML, offre comunque la massima libertà nell'organizzazione dell'interfaccia e consente di realizzare la personalizzazione di un sistema anche complesso in tempi veramente contenuti.

La configurazione di HSYCO è basata esclusivamente su normali file di testo, che possono essere facilmente modificati con qualunque editor di testi su PC; non è quindi richiesto alcun software dedicato per l'attività di configurazione.

E' però anche possibile creare skin alternativi allo skin standard, o lavorare direttamente a livello di HTML ed eventualmente JavaScript per avere la massima flessibilità nella creazione dell'interfaccia utente.

Nel seguito del capitolo verranno descritti i dettagli della configurazione basata sullo skin standard, per poi approfondire alcuni aspetti della configurazione più a basso livello e del sistema di gestione degli skin.

Il Database di Configurazione

Il file **systemtopo.txt** contiene l'elenco completo di tutti i data point degli I/O Server da controllare direttamente dall'interfaccia utente. Contiene inoltre i nome delle zone **Wi-Fi**.

Alcuni I/O Server supportano la funzione di auto-rilevamento dei dispositivi collegati; in questo modo tutti i data point controllabili direttamente dall'interfaccia utente possono essere aggiunti automaticamente al file systemtopo.txt.

Consultare le Application Notes per configurare opportunamente gli I/O Server.

Inoltre, systemtopo.txt può essere modificato in qualunque momento; le modifiche vengono automaticamente applicate entro pochi secondi senza che sia necessario il riavvio di HSYCO. Le pagine Web vengono ricaricate automaticamente in seguito ad ogni modifica di questo file.

Il formato del file è estremamente semplice, suddiviso in sezioni contenenti l'elenco dei dispositivi presenti nel sistema. Ad esempio:

```
(location)
1p : FIRST FLOOR
2p : SECOND FLOOR

(devices)
k.light.14 : LIGHT ; LIGHT ;
k.light.21 : LIGHT ; DIMMER ; DIMMER
k.autom.14 : AUTOM ; VSHUT ;
```

Ogni sezione inizia con una riga in cui il nome della sezione è racchiuso tra parentesi, seguito dall'elenco dei dispositivi relativi alla sezione. Ogni riga inizia con l'id che identifica il dispositivo specifico, seguito da ":" e dall'elenco dei parametri separati da ";". Il file viene letto ignorando le righe vuote e gli spazi non significativi all'interno delle righe.

Eventuali errori nel contenuto del file vengono riportati nel log. In caso di errore generico di sintassi il messaggio di errore è:

systemtopo.txt Parser: Syntax Error, line: nnn

Viene sempre riportato il numero della riga del file del primo errore incontrato. In caso di errore il file non viene interpretato nemmeno parzialmente, di conseguenza l'interfaccia Web non può funzionare. Il contenuto di systemtopo.txt deve quindi essere formalmente e sostanzialmente corretto per consentire il funzionamento dell'interfaccia Web.

Le sezioni del file possono essere omesse se non ci sono dispositivi associati. In realtà un file systemtopo.txt completamente vuoto è a tutti gli effetti un file corretto, e può avere senso nel caso HSYCO sia utilizzato solo per funzioni di videosorveglianza, di gestione Squeezebox o come telecomando IR virtuale. Ogni sezione deve comparire una sola volta nel file, ripetere la riga di inizio di una sezione già presente costituisce un errore.

(location)

Deve contenere una riga per ciascuno degli id degli Access Point elencati nel parametro **LocationBases** nel file di configurazione hsyco.ini. Il formato è:

id : <nome zona>

cioè l'id della zona come in LocationBases, seguito da ":" e dal nome descrittivo della zona. Il nome descrittivo della zona viene visualizzato nel display in alto a destra, ed è quindi obbligatorio.

Esempio:

open : OPEN SPACE

(devices)

Contiene una riga per ogni data point dell'IO Server utilizzato nell'interfaccia Web, con il seguente formato:

id : <funzione>; <tipo>; ; <opzioni>; <descrizione>

id è l'identificativo del data point, una stringa separata da un punto senza spazi. Consultare le Application Notes per i vari formati dei data point.

Il campo **<funzione>** può assumere i valori LIGHT o AUTOM, in base alla configurazione⁹ dell'attuatore associato all'indirizzo bus.

Il campo **<tipo>** indica il tipo di funzione del comando, ai fini della visualizzazione grafica Web. Può assumere uno dei seguenti valori:

STD, LIGHT, DIMMER, VSHUT, HSHUT, LOCK, ONOFF, SCENE, corrispondenti alla grafica illustrata nella tabella seguente¹⁰, con lo skin *blue* standard.

Object Type							
Status	LIGHT	STD	VSHUT	HSHUT	LOCK	ONOFF	SCENE
ERROR							
OFF							
ON							
UP							
DOWN							
GOING UP							
GOING DOWN							
UNKNOWN							

Il campo **** è opzionale. Può essere omissso, quindi una riga come la seguente ha un formato comunque valido:

⁹ Per gli attuatori configurati in modalità interblocco deve essere impostato AUTOM, altrimenti LIGHT.

¹⁰ Il tipo DIMMER non è rappresentato in tabella, perché uguale a LIGHT, con la sola differenza che la barra orizzontale assume una lunghezza diversa in funzione del livello impostato, a passi del 20%, 40%, 60% 80% e 100% di intensità.

k.12 : AUTOM; VSHUT

Il campo può essere utilizzato per associare una immagine o una telecamera al dispositivo¹¹. Per le immagini è sufficiente utilizzare il nome, completo di estensione, di un file presente nella directory *www/img*. Per associare una telecamera deve invece essere utilizzato il formato *cam:N* dove N è l'indice a partire da 1 della telecamera, nell'ordine con cui le telecamere sono elencate nel parametro **Cameras** in *hsyco.ini*.

Il campo **<opzioni>** non è obbligatorio, quindi può essere omesso. Può assumere il valore *NOCLICK* per disattivare a livello di interfaccia Web la possibilità di modificare lo stato, cioè di inviare il comando di gruppo. Il relativo bottone consentirà solo la visualizzazione dello stato dell'indirizzo di gruppo.

Il campo **<descrizione>** è un commento. Non è rilevante ai fini del funzionamento di HSYCO e può essere omesso, ma viene utilizzato nel Project Editor quando viene visualizzata la lista dei dispositivi associati agli oggetti grafici.

Esempio:

k.11 : LIGHT; LIGHT; entrance.jpg; LUCE INGRESSO

¹¹ Cliccando sul bottone o passando sull'area del bottone con il mouse viene mostrata un'immagine statica o il video della telecamera all'interno di un piccolo pop-up di dimensioni 120x80 pixel di fianco al bottone.

Le Pagine Web

Le pagine Web vengono realizzate usando un semplice formato descrittivo testuale che permette di definire pagine multiple e link per la navigazione in un unico file, chiamato ***index.hsm***. E' possibile avere più file index.hsm, ciascuno nella propria directory sotto la directory www. In questo modo è possibile implementare un diverso layout per la navigazione, funzioni e grafica per ciascun utente o gruppo di utenti.

Non è possibile avere più file di descrizione dell'interfaccia Web all'interno della stessa directory.

Quando HSYCO rileva qualunque modifica ai file presenti in www e nelle sotto-directory, forza automaticamente il refresh della pagina Web sui browser collegati, in modo da rendere immediatamente visibili le modifiche effettuate.

Per ogni sotto-directory in www in cui sia presente il file index.hsm sarà possibile accedere alla pagina con l'url:

https://<nome server>/<URLKey> /<nome sotto-directory>

E' possibile creare e modificare le pagine manualmente editando i file index.hsm, ma nella maggior parte dei casi è preferibile l'utilizzo del Project Editor.

Il Project Editor

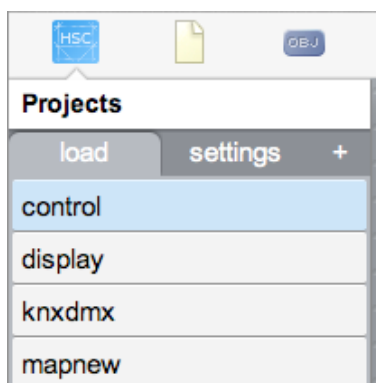
Il Project Editor, che fa parte delle applicazioni del Manager, è un potente editor grafico Web-based utile per creare l'interfaccia Web di HSYCO.

Per accedere al Manager, digitare il seguente URL nel browser Web:

```
https://192.168.0.50/hsycoserver/manager
```

e premere sull'icona del Project Editor.

Sono presenti tre differenti icone nella parte in alto a sinistra della pagina: Projects, Pages e Objects.



La sezione Projects consente di effettuare l'upload di un progetto, modificare i suoi parametri, duplicare, eliminare o creare un nuovo progetto.

La sezione Pages consente di creare e modificare pagine o pop-up che fanno parte di un progetto.

Infine, la sezione Objects consente di gestire i singoli oggetti all'interno di una pagina.

Creare un Nuovo Progetto

Per creare un nuovo progetto, premere sul simbolo + , che mostra una finestra contenente i campi da compilare per le impostazioni del progetto.

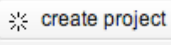
E' obbligatorio assegnare un nome univoco al progetto, che viene utilizzato come sotto-directory di www e come URL per accedere al progetto dal browser Web.

Lo skin "blue" viene selezionato automaticamente. Questo è lo skin standard in HSYCO 3.0. Se sono presenti altri skin personalizzati, essi compariranno all'interno del selettore dello Skin.

Language consente di scegliere la lingua del progetto tra Italian, English e French, per lo skin blue. La lingua scelta verrà utilizzata per tutti i messaggi di testo all'interno del progetto creato.

Header è un campo di testo vuoto che corrisponde all'header della pagina Web (il testo che solitamente compare tra i preferiti e in alto alla finestra del browser).

Le opzioni sulle dimensioni del progetto verranno descritte nel capitolo seguente.

Una volta compilati questi campi, premere l'icona  per confermare la creazione del nuovo progetto. Il Project Editor crea una directory per questo progetto con la struttura di base del file index.hsm.

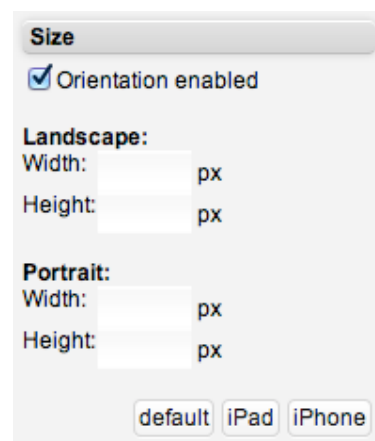


Layout e Coordinate delle Pagine

Quando viene creato un progetto, è necessario scegliere la dimensione per adattare il layout dello schermo dei dispositivi con cui accedere all'interfaccia di HSYCO.

All'interno della sezione Size, premere i bottoni iPad o iPhone per impostare la dimensione predefinita che si adatta a questi dispositivi, così come a tanti altri, o scegliere una dimensione personalizzata digitando larghezza e altezza in pixel. E' possibile impostare anche una larghezza maggiore, da utilizzare su desktop o computer portatili, oppure un'altezza maggiore, che ben si adatta a dispositivi come iPhone o iPod touch o altri dispositivi mobili.

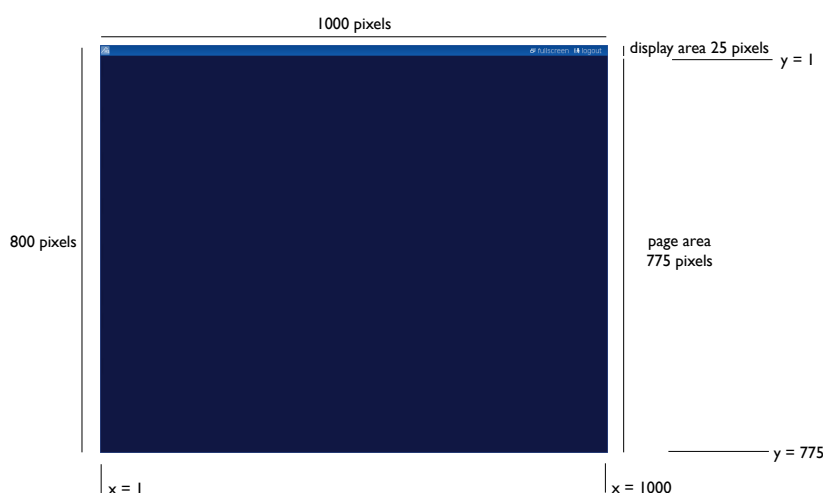
E' anche possibile abilitare il supporto per l'orientamento. Impostando il flag su Orientation enabled sui tablet e dispositivi mobili che supportano la rotazione dello schermo basata sull'orientamento del dispositivo, è possibile definire pagine distinte all'interno dello stesso progetto per mostrare diversamente il caso in cui il dispositivo sia orientato verticalmente o orizzontalmente. In questi casi, è anche possibile definire layout completamente diversi



per le pagine con orientamento orizzontale (landscape) o verticale (portrait), in particolare con layout portrait ottimizzato per iPhone e landscape per iPad o personal computer.

Se il dispositivo non supporta l'orientamento, verrà utilizzato il formato in versione landscape per mostrare l'interfaccia Web del progetto.

Nella seguente immagine vengono mostrati il layout e le dimensioni dell'area di una pagina, basati ad esempio su una dimensione della finestra di 1000x800 pixel.



E' possibile utilizzare diversi formati per il campo della posizione degli oggetti Web:

- **coordinate X-Y in pixel.** Per esempio, x100y200 posiziona l'origine dell'oggetto al pixel 100 in orizzontale e 200 verticale, a partire dall'angolo in alto a sinistra dell'area del display
- **coordinate in righe-colonne.** Per esempio, 23 o r2c3 posiziona l'oggetto alla riga 2, e colonna 3 in base alla griglia di riferimento. E' possibile usare una rappresentazione abbreviata a due cifre per righe e colonne, o usare le lettere R e C (maiuscole o minuscole). Ad esempio, sia r1c1 che 11 rappresentano la posizione in alto a sinistra della griglia. La griglia di riferimento dipende dalla dimensione dell'oggetto che la usa.

Modificare un Progetto

Cliccando sulle impostazioni di un progetto esistente, si accede a una serie di parametri aggiuntivi che condizionano il comportamento generale dell'interfaccia Web del progetto.

Il parametro Device Image abilita o disabilita il pop-up con l'immagine della telecamera o l'immagine associata ai dispositivi nell'interfaccia Web, così come configurata nel file systemtopo.txt.

La modalità Kiosk viene utilizzata per rimuovere la barra del menu e i bordi, e rappresenta la soluzione ideale per applicazioni kiosk o di digital signage in cui si ha l'esigenza di ottenere display full-screen delle pagine. Selezionando la modalità "lock" vengono disabilitate anche le funzioni di navigazione.


Scale imposta un fattore di scala prefissato per ridurre o aumentare la dimensione dell'interfaccia di HSYCO. Valori maggiori di 1.0 causano un ingrandimento della pagina rispetto alle dimensioni originali; numeri minori di 1.0 causano una riduzione delle dimensioni originali.

The screenshot shows a vertical list of configuration settings in a light gray interface. Each setting has a title bar and a control element:


- Device Image:** A dropdown menu currently set to "enabled".
- Kiosk:** A dropdown menu currently set to "disabled".
- Scale:** An empty text input field.
- Locked:** A section header with a checkbox labeled "Project locked" which is currently unchecked.
- Background:** A text input field with a three-dot menu icon to its right.
- Camera List:** A text input field containing the text "Front", "Back", "Walk" with a three-dot menu icon to its right.
- Camera Overlay:** A text input field with a three-dot menu icon to its right.
- Camera Grid Headers:** A text input field with a three-dot menu icon to its right.

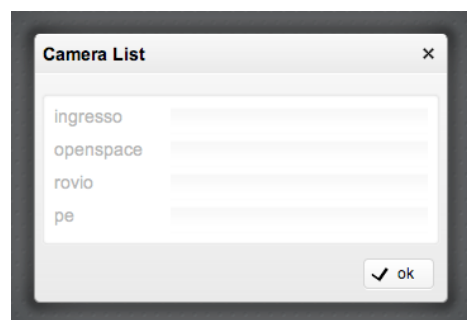
Selezionando “Project locked” qualsiasi azione o comando derivante dalla pressione di bottoni all’interno dell’interfaccia Web del progetto viene disabilitata.

Background consente di selezionare una immagine come sfondo per tutte le pagine del progetto.

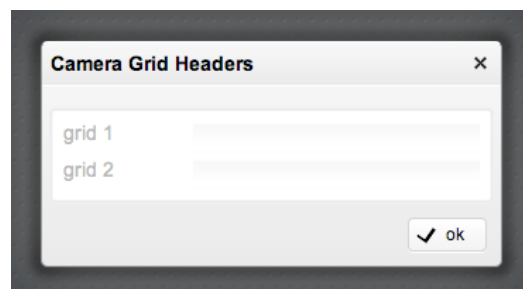
Premendo l'icona  all’interno della sezione Camera List, viene mostrato un pop-up in cui è possibile inserire una descrizione per ognuna delle telecamere elencate nel file hsyco.ini.

E’ possibile lasciare i campi vuoti se non si vuole mostrare alcuna telecamera nel progetto.

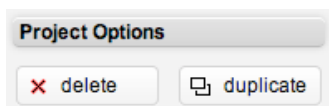
Premendo l'icona  all’interno della sezione Camera Overlay, viene mostrato un pop-up in cui è possibile specificare un’immagine-maschera in sovra-impressione per ogni telecamera. E’ opzionale, e se non viene specificata HSYCO utilizzerà l’immagine standard per telecamere PTZ.



Infine, come per la sezione Camera List, Camera Grid consente di inserire una descrizione per le griglie di telecamere dichiarate in hsyco.ini.



Infine, è possibile eliminare o duplicare un progetto cliccando rispettivamente sui bottoni delete o duplicate.



Menu, Pagine e Pop-Up

Esistono tre tipi di pagine all'interno dell'interfaccia Web: i menu, le pagine e i pop-up.

Per creare una nuova pagina cliccare su Pages e in seguito sul simbolo +.

I menu sono come pagine tradizionali, ma la pagina di menu è anche la home page del progetto. Può essere presente un solo menu per progetto, o due nel caso vengano utilizzate entrambe le versioni landscape e portrait con la modalità dell'orientamento abilitata.

Cliccare su menu per creare una pagina di menu, selezionare l'orientamento se abilitato. Non sono presenti altri parametri per i menu.

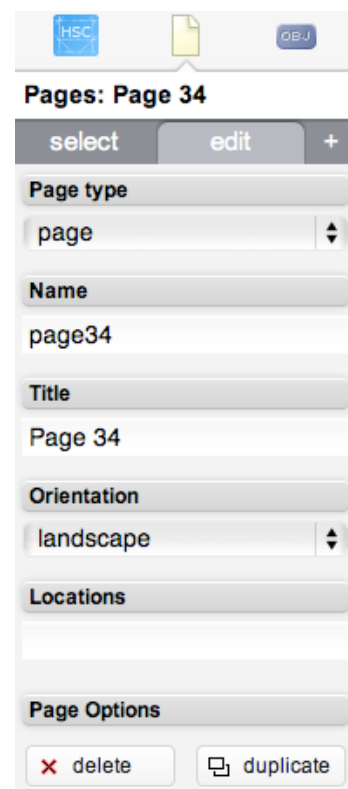
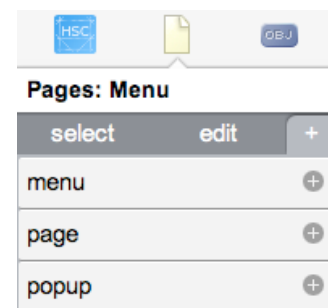
Cliccare su page per creare una nuova pagina, o selezionare una pagina tra quelle già create per apportare modifiche.

Quanto viene creata una nuova pagina, il Project Editor assegna automaticamente ad essa un nome (che rappresenta l'ID della pagina) e un titolo di default. E' possibile modificare il nome della pagina e il titolo a piacere.

Selezionare l'orientamento, se abilitato.

Il campo Location consente di associare una pagina a uno o più ID di location, così come definiti nel parametro LocationBases in hsyco.ini. Grazie all'associazione tra pagine e location, è possibile cliccare sul nome della location mostrato nell'area in alto a destra della barra di menu per spostarsi tra una pagina e l'altra associate a quell'area.


Una pagina può essere eliminata o duplicata ad una pagina identica, assegnando un nome diverso (ID).

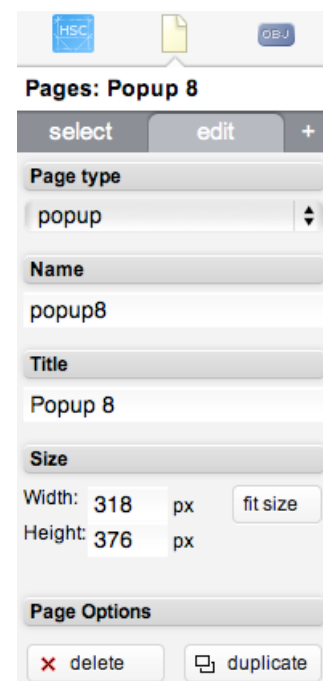


Cliccare su pop-up per creare un nuovo pop-up.


Un pop-up è una pagina più piccola che compare sopra una pagina esistente o ad un altro pop-up.

Allo stesso modo di una pagina, è possibile modificare il nome e il titolo di un pop-up e impostarne anche larghezza e altezza in pixel.

Il bottone  adatta automaticamente la dimensione del pop-up in base agli oggetti aggiunti all'interno del pop-up.



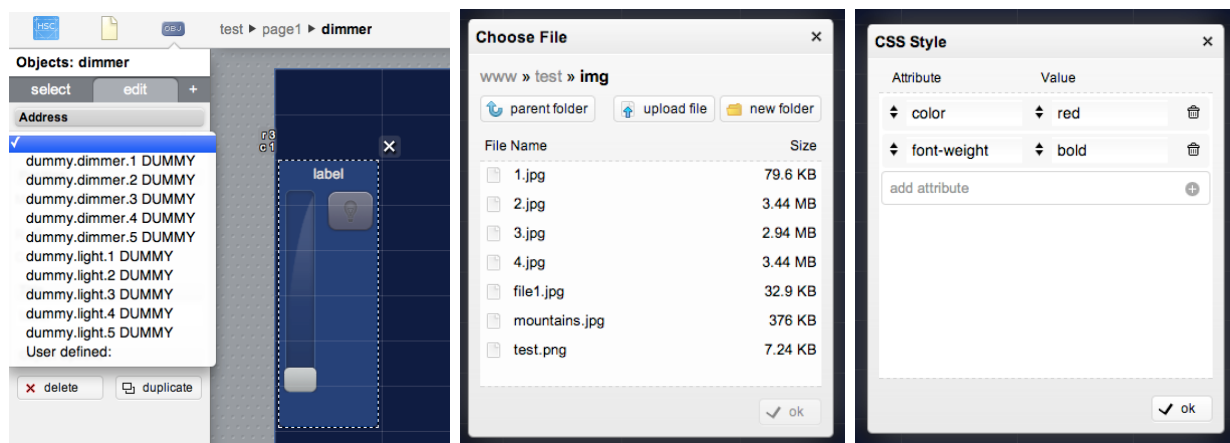
Gli Oggetti di una Pagina

E' possibile aggiungere e modificare la grafica e gestire gli oggetti all'interno di una pagina. Cliccando sul bottone  verrà mostrata una lista con tutti gli oggetti contenuti nella pagina. Cliccando su qualsiasi oggetto della lista esso viene evidenziato all'interno della pagina, e selezionando la sezione edit è possibile modificarne i suoi attributi.

Cliccare sul simbolo + per aggiungere un nuovo oggetto. Selezionare un oggetto dalla lista di tutti gli oggetti disponibili e trascinarlo nella posizione desiderata. Infine modificare i suoi attributi.



La sezione edit mostra tutte le proprietà dell'oggetto selezionato. Per alcuni campi è possibile scegliere un attributo da una lista di valori, ad esempio l'indirizzo degli oggetti button. Per altri, come gli oggetti image, si aprono finestre guidate, wizard, che consentono di scegliere le immagini tra i file esistenti. Infine, per gli oggetti di cui è possibile modificare lo stile CSS, è possibile selezionare gli attributi all'interno di una finestra di stile guidata.



Cliccando sopra le coordinate dell'oggetto si passa dalla modalità di spostamento a righe/colonne o a pixel e viceversa.

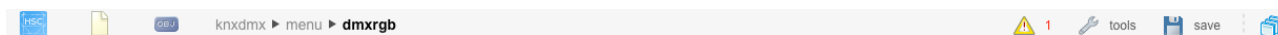


Selezionando un oggetto all'interno della pagina è anche possibile modificare l'ordine di visualizzazione dell'oggetto utilizzando i bottoni move up e move down all'interno degli oggetti della lista.



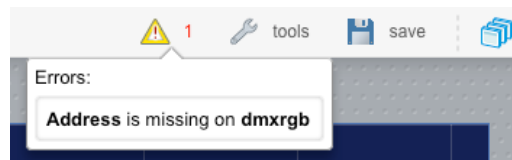
Gli oggetti più in alto nella lista appariranno sopra gli altri nella pagina Web.

Una barra degli strumenti in alto mostra delle funzioni specifiche del Project Editor.

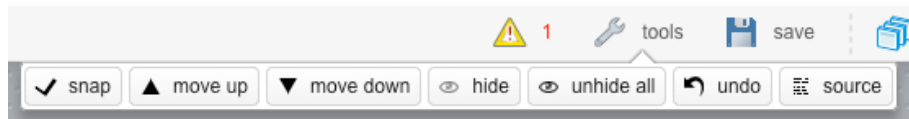


Vicino alle icone dei progetti, pagine e oggetti, viene mostrata la gerarchia di navigazione, che specifica il nome del progetto, il nome della pagina corrente, e l'oggetto selezionato. E' possibile cliccare su ognuno di questi elementi per spostarsi rapidamente tra la lista di oggetti o la lista delle pagine.

Se sono presenti errori verrà mostrata un'icona di avviso (ad esempio se mancano alcune dichiarazioni di parametri degli oggetti). Cliccando sull'icona di avviso verrà mostrato il messaggio di errore evidenziando direttamente l'oggetto in questione e mostrando l'attributo da correggere.



L'icona tools mostra un menu che fornisce utili funzioni per la gestione degli oggetti aggiunti nella pagina.



Se snap è abilitato, gli oggetti possono essere spostati all'interno della pagina allineandosi automaticamente alla griglia.

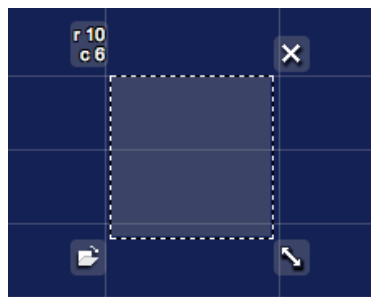
Il bottone undo consente di annullare i cambiamenti effettuati fino all'ultima modifica salvata.

Infine, i bottoni hide e unhide all consentono rispettivamente di nascondere un oggetto e di far ricomparire l'oggetto o gli oggetti precedentemente nascosti.

Container

Un container non è un vero e proprio oggetto. Viene utilizzato per raggruppare un insieme di altri oggetti. La posizione di tutti gli oggetti all'interno del container è relativa alla posizione del container stesso. Possono esistere anche container all'interno di un altro container.

Per creare un container, cliccare sul simbolo + e scegliere l'oggetto container dalla lista di oggetti.



L'oggetto container comparirà all'interno della pagina. E' possibile modificare le dimensioni cliccando nell'angolo in basso a destra e trascinare la finestra del

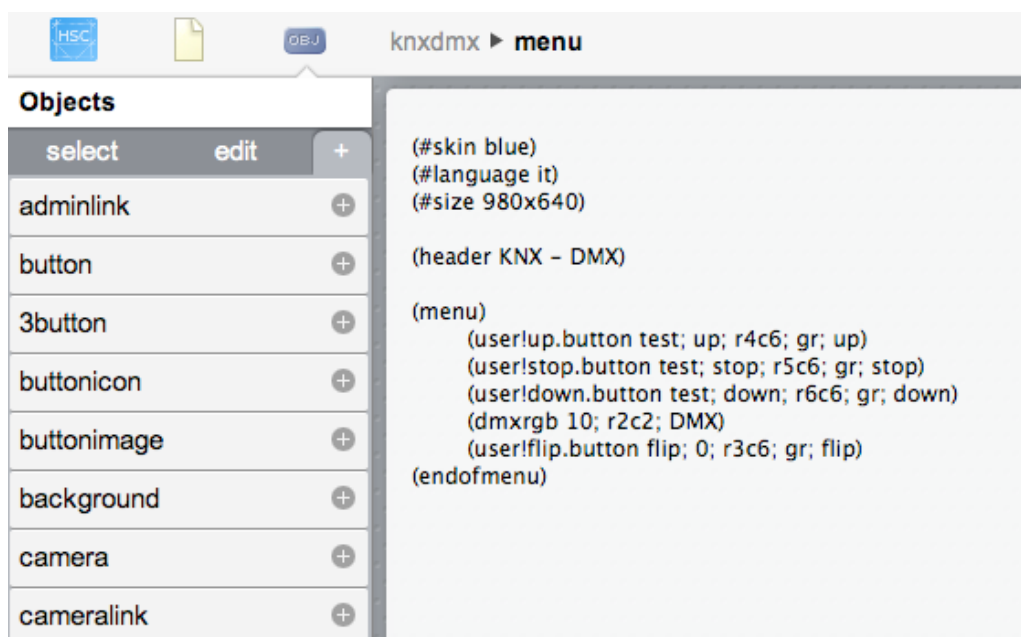
container fino alla posizione desiderata, oppure aprire il container per modificare gli oggetti contenuti all'interno cliccando nell'angolo in basso a sinistra.



Una volta apportate le modifiche agli oggetti contenuti al suo interno, chiudere il container cliccando nuovamente nell'angolo in basso a sinistra.

Il file sorgente del progetto index.hsm

Cliccando sul bottone source nella barra degli strumenti, il Project Editor consente di visualizzare il codice sorgente del file index.hsm.



Attraverso il File Manager è possibile aprirlo e modificare direttamente il contenuto del progetto. Se il progetto era stato aperto, ritornando al Project Editor verrà richiesto di caricare le modifiche effettuate.

Objects Reference

Oggetti

Il formato del file index.hsm è basato su oggetti grafici, la cui rappresentazione generica è:

(tipo_oggetto <parametro1>;<parametroN>)

Eventuali errori nel contenuto del file vengono riportati nel log. In caso di errore di sintassi il messaggio di errore è:

Page Parser: Syntax Error, line: nnn

Viene sempre riportato il numero della riga del file del primo errore incontrato. In caso di errore il file non viene interpretato nemmeno parzialmente, di conseguenza l'interfaccia Web non può funzionare.

Ogni tipo di oggetto grafico può richiedere o meno uno o più parametri, separati dal carattere “;”. Vengono ignorate le righe vuote e gli spazi non significativi all'interno delle righe.

Le modifiche effettuate al file index.hsm vengono immediatamente rese disponibili dal server Web di HSYCO.

E' prevista la possibilità di inserire righe di commento, che vengono ignorate ai fini della configurazione. Una riga di commento è identificata dal carattere “#” a inizio riga.

Per i nomi degli oggetti in index.hsm, le lettere maiuscole e minuscole sono significative; tutti gli oggetti definiti nello skin standard hanno nomi costituiti da lettere minuscole.

Per alcuni oggetti è possibile definire una **versione identificata** che consente la modifica dinamica degli attributi testo, visibilità, colore e immagine con il metodo Java `uiSet()` o con l'azione `UISET`.

!id

L'oggetto è identificato da una stringa `<id>` che deve essere univoca all'interno del file `index.hsm`. La presenza di più oggetti con lo stesso id può causare effetti indesiderati, dipendenti dal tipo di Web Browser utilizzato.

Rispetto alla versione non identificata, il nome dell'oggetto è immediatamente seguito da un punto esclamativo e da un `<id>` univoco.

Ad esempio:

```
(text!miotesto r1c1; Questo è il mio oggetto testo identificato)
```

Direttive Generali

(#skin <name>)

Questa direttiva, obbligatoria, deve essere posizionata all'inizio del file `index.hsm`, prima dell'oggetto (header). Lo skin standard fornito con HSYCO è "blue".

Parametri:

`<skin name>` - nome dello skin utilizzato per l'interfaccia Web.

(#language <language id>)

Questa direttiva, se presente, deve essere posizionata prima dell'oggetto (header) e definisce la lingua utilizzata per tutti i messaggi di testo dell'interfaccia Web.

Parametri:

`<language id>` - lingua utilizzata per l'interfaccia Web (it | en | fr).

(#size <page size>)

Imposta la dimensione della pagina di un progetto, e deve essere posizionata prima dell'oggetto (header).

Per i progetti che supportano l'orientamento la sintassi è:

```
(#size <landscape size>; <portrait size>)
```

La direttiva (#size) può anche essere utilizzata per impostare la dimensione dei pop-up. In questo caso deve essere scritta subito dopo l'oggetto (popup), prima di qualunque altro oggetto definito all'interno del pop-up.

Per esempio:

```
(#size 1200x800)
```

imposta una dimensione di 1200 pixel di larghezza e 800 pixel di altezza.

```
(#size r12c18)
```

imposta una dimensione a 12 righe e 18 colonne, basata su una griglia di riferimento utilizzata per i bottoni di dimensioni standard.

Parametri:

<page size> - dimensioni in pixel o a righe e colonne.

(#scale <factor>)

Imposta un fattore di scala per dimensionare l'intera area della pagina rispetto alla sua dimensione originale. Questa direttiva è supportata solo su Safari per iOS.

Parametri:

<factor> - fattore di scala dell'interfaccia. E' un numero decimale maggiore di zero: numeri maggiori di 1.0 causano un ingrandimento della pagina rispetto alle dimensioni originali; numeri minori di 1.0 causano una riduzione delle dimensioni.

(#kiosk <mode>)

consente di visualizzare le pagine a schermo pieno, senza la barra di menù in alto ed il bordo. (#kiosk lock) disabilita i tasti di tipo link e la navigazione ad altre pagine.

Parametri:

<mode> - parametro opzionale. Se impostato a lock disabilita i tasti di tipo link e la navigazione.

(#locked <mode>)

Disabilita qualsiasi comando utente del progetto. Gli utenti possono navigare tra le pagine e vedere lo stato corrente, ma tutti i comandi sono bloccati.

Parametri:

<mode> - impostato a "true" disabilita l'esecuzione di comandi.

(#deviceimage disable)

Abilita o disabilita la visualizzazione di pop-up contenenti l'immagine della telecamera o l'immagine associata ai dispositivi, così come configurati nel file systemtopo.txt.

(#location <list>)

Consente di associare una pagina ad uno o più location id, come definiti dal parametro *LocationBases* in hsyco.ini. Con questa associazione, cliccando sul nome della zona visualizzato in alto a destra, vengono visualizzate ciclicamente tutte le pagine associate alla zona.

Deve essere specificata subito dopo l'oggetto (page).

Parametri:

<list> - lista di location id, separati da punto e virgola, come definiti dal parametro LocationBases in hsyco.ini.

(#cameralist <list>)

Questa direttiva è necessaria per definire il nome visualizzato per ciascuna telecamera, e per disabilitare l'accesso ad una o più telecamere.

Se si desidera impedire l'accesso a una o più telecamere da uno specifico index.hsm, è possibile impostare tale telecamera con un nome vuoto, "", o i nomi riservati "null" o "empty". Queste telecamere saranno saltate durante la funzione di rotazione delle telecamere, o quando si passa manualmente da una telecamera all'altra.

Parametri:

<list> - lista dei nomi delle telecamere, elencati nell'ordine con cui gli id delle telecamere sono riportati nel parametro Cameras in hsyco.ini. I nomi devono essere racchiusi tra doppi apici e separati da virgola.

(#cameragridlist <list>)

Questa direttiva è necessaria per definire il nome visualizzato per ciascuna griglia di telecamere.

Parametri:

<list> - lista dei nomi dei gruppi di telecamere, elencati in ordine a partire dal gruppo 1, come definiti nei parametri CameraGrid.id in hsyco.ini. I nomi devono essere racchiusi tra doppi apici e separati da virgola.

(#cameraoverlay <overlay list>)

Questa direttiva consente di dichiarare immagini da sovrapporre alla visualizzazione delle telecamere. E' opzionale, se non specificato HSYCO utilizzerà l'overlay standard per le sole telecamere PTZ.

Parametri:

<overlay list> - lista del tipo di overlay per ciascuna telecamera definita in hsyco.ini, con gli elementi separati da virgola:

“default” - overlay di default per telecamere PTZ

“” - overlay di default per telecamere PTZ

“null” - nessun overlay

“nomefile.png” - file personalizzato, nella directory www/img.

(#include <file name>)

Consente di includere nel file index.hsm il contenuto di un altro file presente nella stessa directory. Il nome del file da includere deve avere estensione .hsm; l'estensione non deve essere specificata nella direttiva (#include).

E' possibile inserire più (#include) in un file, in qualunque punto del file.

Parametri:

<name file> - nome del file da includere.

Menù Principale

(header <title>)

L'header corrisponde all'header della pagina Web, tipicamente visualizzato dai Web Browser nella barra di stato in alto.

Questo oggetto deve essere obbligatoriamente presente come primo oggetto nel file index.hsm, subito dopo la dichiarazione delle direttive generali.

Parametri:

<title> - titolo della pagina.

(menu)

Definisce l'inizio della pagina del menu principale, visualizzata dopo il login. E' possibile specificare una sola pagina di menu all'interno del file index.hsm.

(menu#landscape)

Versione con orientamento orizzontale della pagina di menu. Utilizzando uno skin a dimensioni variabili, è possibile creare due versioni distinte del menu, per la visualizzazione in formato orizzontale (base maggiore dell'altezza della pagina) o verticale (base minore dell'altezza). La pagina visualizzata sarà scelta automaticamente in base all'orientamento fisico del dispositivo.

(menu#portrait)

Versione con orientamento verticale della pagina di menu.

(endofmenu)

Indica la fine della pagina di menu principale.

Pagine

(link <pos>; <color>; <page>; <label>)

!id

(linkmini <pos>; <color>; <page>; <label>)

!id

(linkmicro <pos>; <color>; <page>; <label>)

!id

Questo oggetto può essere contenuto all'interno della pagina di menu o in qualunque altra pagina. Corrisponde ad un bottone di accesso ad un'altra pagina.

Per gli oggetti (link), (linkmini) e (linkmicro) esiste la versione identificata:

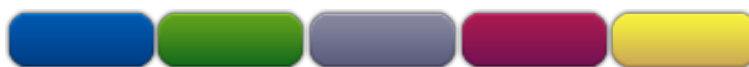
(link!id <pos>; <color>; <page>; <label>)

(linkmini!id <pos>; <color>; <page>; <label>)

(linkmicro!id <pos>; <color>; <page>; <label>)



link



linkmini



linkmicro

Parametri:

<pos> - posizione del bottone link all'interno della pagina. E' possibile utilizzare il formato coordinate X-Y in pixel o in righe-colonne

<color> - colore del bottone. Può assumere i valori: *b* - blu; *g* - verde; *gr* - grigio; *r* - rosso; *y* - giallo

<page> - nome identificativo della pagina cui il bottone link si riferisce. E' possibile specificare un URL assoluto o relativo, che verrà caricato in una nuova pagina. Se la pagina caricata è una pagina di menu di HSYCO, attraverso il link [< back] dell'interfaccia sarà possibile visualizzare la pagina precedente

<text> - testo presentato all'interno del bottone link. Esso può contenere anche istruzioni HTML.

(dlink <pos>; <color>; <page>; <label>)

!id

Analogo all'oggetto (link), con l'unica differenza che la grafica visualizzata è quella di un mini-display invece del normale bottone.

Per l'oggetto (dlink) esiste la versione identificata:

(dlink!id <pos>; <color>; <page>; <label>)



Parametri:

<pos> - posizione del bottone dlink all'interno della pagina. E' possibile utilizzare il formato coordinate X-Y in pixel o in righe-colonne

<color> - colore del bottone. Può assumere i valori: *b* - blu; *r* - rosso

<page> - nome identificativo della pagina cui il bottone dlink si riferisce

<text> - testo presentato all'interno del bottone dlink. Esso può contenere anche istruzioni HTML.

(page <id>; <description>; <protection>)

(page#landscape <id>; <description>; <protection>)

(page#portrait <id>; <description>; <protection>)

Questo oggetto rappresenta una pagina generica.

page#landscape definisce la versione della pagina specifica per l'orientamento orizzontale; page#portrait definisce la pagina con orientamento verticale. Alcuni nomi sono riservati e non possono essere utilizzati, nemmeno come inizio di una stringa più lunga. I nomi riservati sono: *menu*, *temp*, *music*, *cameras*, *timer*, *logout*. Mentre è normale avere più oggetti link che puntano alla stessa pagina, avere più oggetti (page) con lo stesso nome costituisce un errore, che può portare ad un funzionamento non prevedibile del Web Browser, anche se questo tipo di errore semantico non viene rilevato dal server Web di HSYCO.

Parametri:

<id> - id della pagina

<description> - titolo della pagina, che appare nella zona centrale in basso del display quando la pagina viene visualizzata

<protection> - parametro opzionale. Può essere impostato a "pin" o "puk". Consente di proteggere l'accesso alla pagina forzando l'autenticazione con PIN o con PIN e PUK¹².

E' importante sottolineare che l'autenticazione con PIN o PUK a livello di pagina è una protezione realizzata lato client, e potrebbe essere aggirata da una persona dotata delle conoscenze opportune, per consentire comunque l'accesso a pagine protette. E' comunque utile per evitare l'accesso accidentale a pagine specifiche.

¹² L'autenticazione a livello di pagina è richiesta solo se sono trascorsi almeno 60 secondi dall'ultima autenticazione.

Utilizzare il sistema di protezione basato sulle Access Control List lato server per una maggiore protezione dei comandi critici.

(popup <id>; <description>)

Questo oggetto è simile all'oggetto (page).

Mostra l'oggetto come un pop-up che compare al di sopra della pagina originale, tale oggetto può essere usato solo con gli skin di dimensioni variabili.

E' possibile utilizzare la direttiva (#size) per impostare le dimensioni di ciascun pop-up. Deve essere scritta subito dopo l'oggetto (popup), prima di qualunque altro oggetto definito all'interno del pop-up.

Se (#size) non è specificata, il pop-up avrà dimensioni standard, dipendenti dallo skin utilizzato.

Parametri:

<id> - id del pop-up

<description> - titolo del pop-up, che appare nella zona centrale in basso del display quando la pagina viene visualizzata.

(endofpage)

Definisce la fine della pagina di (page) o (popup).

(container <pos>; <visibility>)

!id

Questo oggetto viene utilizzato per raggruppare altri oggetti al suo interno. La posizione di tutti gli oggetti interni al container è relativa rispetto a quella del container stesso. I container possono anche essere annidati uno dentro l'altro.

Per l'oggetto (container) esiste la versione identificata:

(container!id <pos>; <visibility>).

Parametri:

- <pos> - posizione del container all'interno della pagina. E' possibile utilizzare il formato coordinate X-Y in pixel o in righe-colonne
- <visibility> - opzionale. Impostato a "hidden" rende il container invisibile quando la pagina viene caricata.

(endofcontainer)

Definisce la fine del container.

(selector <file>; <pos>; <width>; <height>; <container id>; <group>; <text>; <css>)

Il selector consente di creare un oggetto grafico che contiene al suo interno diverse pagine secondo una navigazione a tab, anche più flessibile di quest'ultima. L'aspetto grafico di un selector è lo stesso di un oggetto (image) o (imagelink), un'immagine o una stringa di testo.

E' necessario creare due o più container sovrapposti all'interno di una pagina, ognuna con id univoco. Quindi creare un oggetto (selector) per ogni container, e definire un unico gruppo uguale per tutti i selector creati. Se viene premuto un oggetto (selector), il container corrispondente diventerà visibile, mentre tutti gli altri container associati ai selector che fanno parte dello stesso gruppo diventeranno invisibili.

Parametri:

- <file> - nome del file da utilizzare come immagine
- <pos> - posizione. E' possibile utilizzare il formato coordinate X-Y in pixel o in righe-colonne
- <width> - larghezza, definita in pixels
- <height> - altezza, definita in pixel
- <container id> - id del container associato al selector

`<group>` - opzionale. Nome del gruppo che identifica un insieme di selector. Tutti i selector senza nome di gruppo definito si considerano facenti parte dello stesso gruppo

`<text>` - testo opzionale. Può contenere anche istruzioni HTML

`<css>` - stile CSS opzionale.

(background <file>)

!id

Questo oggetto consente di definire un'immagine di sfondo. Il file deve essere presente nella directory `www/img` e deve avere dimensioni corrispondenti allo skin utilizzato, o alla dimensione specificata per gli skin a dimensioni variabili.

(background) deve essere posizionato subito dopo (header) se si vuole impostare uno sfondo comune per tutte le pagine, oppure subito dopo gli oggetti (menu), (page) o (popup) per definire sfondi diversi per ogni pagina.

Parametri:

`<file>` - nome del file da utilizzare come immagine di sfondo, presente nella directory `www/img`.

Testi

(text <pos>; <text>; <css>)

!id

Questo oggetto visualizza un testo generico.

Per l'oggetto (text) esiste la versione identificata:

(text!id <pos>; <text>; <css>).

Parametri:

`<pos>` - posizione del testo all'interno della pagina. E' possibile utilizzare il formato coordinate X-Y in pixel o in righe-colonne

`<text>` - testo visualizzato. Può contenere anche istruzioni HTML

<css> - parametro opzionale. Definisce lo stile CSS del testo.

(marquee <pos>; <width>; <height>; <dir>; <speed>; <text>; <css>) !id

Questo oggetto visualizza un testo generico, come (text) ma con scorrimento del testo visualizzato.

Per l'oggetto (marquee) esiste la versione identificata:

(marquee!id <pos>; <width>; <height>; <dir>; <speed>; <text>; <css>).

Parametri:

<pos> - posizione del testo all'interno della pagina. E' possibile utilizzare il formato coordinate X-Y in pixel o in righe-colonne

<width> - larghezza dell'immagine, specificata in pixel

<height> - altezza dell'immagine, specificata in pixel

<dir> - direzione di scorrimento, può assumere i valori:

left: scorrimento da destra a sinistra

right: scorrimento da sinistra a destra

up: scorrimento verso l'alto

down: scorrimento verso il basso

<speed> - velocità di scorrimento; la velocità minima si ottiene impostando speed a 1, e aumenta usando numeri crescenti

<text> - testo visualizzato. Può contenere anche istruzioni HTML

<css> - parametro opzionale. Definisce lo stile CSS del testo.

Elementi Grafici

(panel <pos>; <width>; <height>; <color>)

!id

Questo oggetto visualizza un pannello di sfondo.

Per l'oggetto (panel) esiste la versione identificata:

(panel!id <pos>; <width>; <height>; <color>)

Parametri:

- <pos> - posizione del panel all'interno della pagina. E' possibile utilizzare il formato coordinate X-Y in pixel o in righe-colonne
- <width> - larghezza dell'immagine, specificata in pixel
- <height> - altezza dell'immagine, specificata in pixel
- <color> - opzionale. Colore del bottone. Se non impostato, il panel di default sarà trasparente. Può assumere i valori: *b* - blu; *g* - verde; *gr* - grigio; *r* - rosso; *y* - giallo.

(hbar <pos>; <width>)

Questo oggetto visualizza una barra orizzontale.

Parametri:

- <pos> - posizione della barra all'interno della pagina. E' possibile utilizzare il formato coordinate X-Y in pixel o in righe-colonne
- <width> - lunghezza della barra, specificata in pixel.

(vbar <pos>; <height>)

Questo oggetto visualizza una barra verticale.

Parametri:

- <pos> - posizione della barra all'interno della pagina. E' possibile utilizzare il formato coordinate X-Y in pixel o in righe-colonne
- <height> - altezza della barra, specificata in pixel.

(image <file>; <pos>; <width>; <height>; <text>; <css>)**!id**

Questo oggetto visualizza l'immagine di un file presente nella directory `www/img` ed un testo opzionale, che può essere formattato e posizionato grazie agli attributi standard CSS. In particolare, l'attributo "top" consente di riposizionare il testo. Se top non è definito, il testo sarà posizionato subito sotto l'immagine.

Definendo un valore positivo o negativo in pixel è possibile spostare il testo rispettivamente in basso o in alto a partire dall'angolo in alto dell'immagine.

Per l'oggetto (image) esiste la versione identificata:

(image!id <file>; <pos>; <width>; <height>; <text>; <css>).

Parametri:

<file> - nome del file da utilizzare come immagine, presente nella directory
www/img

<pos> - posizione dell'immagine all'interno della pagina. E' possibile utilizzare il formato coordinate X-Y in pixel o in righe-colonne

<width> - larghezza dell'immagine, specificata in pixel

<height> - altezza dell'immagine, specificata in pixel

<text> - parametro opzionale. Testo, visualizzato normalmente sotto l'immagine

<css> - parametro opzionale. Definisce lo stile CSS del testo.

(imagelink <file>; <pos>; <width>; <height>; <page>; <text>; <css>) !id

Questo oggetto visualizza l'immagine di un file presente nella directory www/img ed un testo opzionale. L'immagine funziona come un link, cliccando sull'immagine viene visualizzata la pagina indicata nel campo <page>.

Il testo può essere formattato e posizionato grazie agli attributi standard CSS. In particolare, l'attributo "top" consente di riposizionare il testo. Se top non è definito, il testo sarà posizionato subito sotto l'immagine. Definendo un valore positivo o negativo in pixel è possibile spostare il testo rispettivamente in basso o in alto a partire dall'angolo in alto dell'immagine.

Per l'oggetto (imagelink) esiste la versione identificata:

(imagelink!id <file>; <pos>; <width>; <height>; <page>; <text>; <css>).

Parametri:

<file> - nome del file da utilizzare come immagine, presente nella directory
www/img

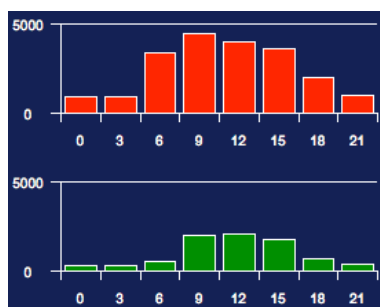
- <pos> - posizione dell'immagine all'interno della pagina. E' possibile utilizzare il formato coordinate X-Y in pixel o in righe-colonne
- <width> - larghezza dell'immagine. Essa deve essere specificata in pixel
- <height> - altezza dell'immagine. Essa deve essere specificata in pixel
- <page> - nome della pagina da visualizzare cliccando sull'immagine. E' possibile specificare un URL assoluto o relativo, che verrà caricato in una nuova pagina. Se la pagina caricata è una pagina di menu di HSYCO, attraverso il link [<back>] dell'interfaccia sarà possibile visualizzare la pagina precedente
- <text> - parametro opzionale. Testo, visualizzato normalmente sotto l'immagine
- <css> - parametro opzionale. Definisce lo stile CSS del testo.

(chart!id <pos>; <width>; <height>; <attributes>)

!id

Questo oggetto consente di visualizzare un grafico. I parametri del grafico e i dati vengono impostati attraverso il metodo Java uiSet() o l'azione UISET in EVENTS.

Per l'oggetto (chart) esiste solo la versione identificata.



I chart che rappresentano una serie di dati possono essere di tipo a barre e a punti. Il chart di tipo “gauge” invece è utilizzato per presentare una barra di lunghezza variabile in funzione di un solo valore.




```
(type=gauge; orientation=horizontal; value=1; valuerange=0,10; bgcolor=black)
(type=gauge; orientation=vertical; value=4; valuerange=0,10; bgcolor=green;
barborder=false)
```

I chart possono avere orientamento verticale, per cui l'asse dei valori è orientato verticalmente, oppure orizzontale.

Parametri:

<pos> - posizione del grafico all'interno della pagina. E' possibile utilizzare il formato coordinate X-Y in pixel o in righe-colonne

<width> - larghezza dell'oggetto (chart) in pixel

<height> - altezza dell'oggetto (chart) in pixel

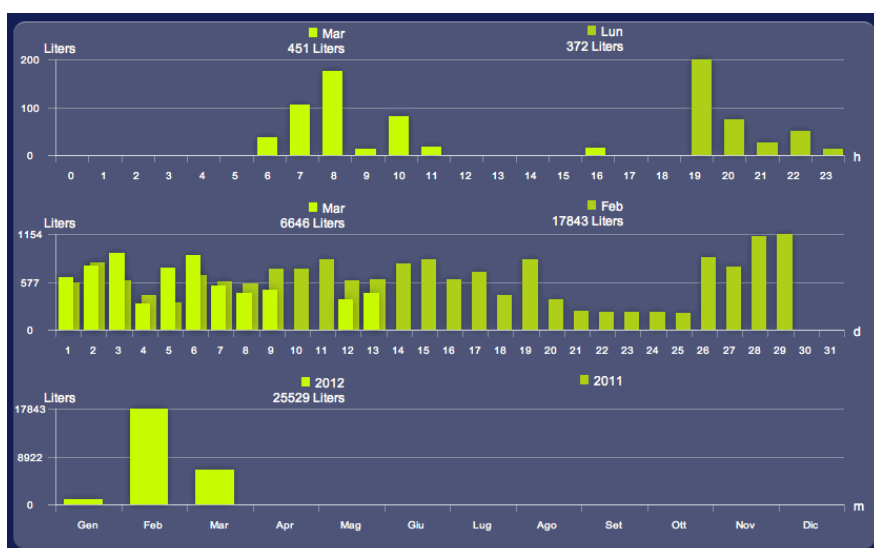
<attributes> - parametro opzionale per impostare gli attributi del grafico.

attributo	valore
visible	<i>true</i> : visibile; <i>false</i> : non visibile
pos	la posizione dell'oggetto
type	tipo di grafico; valori ammessi: <i>bars</i> , <i>points</i> , <i>gauge</i> . Il chart di tipo gauge supporta solo i seguenti attributi: value, orientation, valuerange, bgcolor, barcolor, barborder; l'attributo valuerange è necessario per una corretta rappresentazione grafica del valore
orientation	orientamento verticale o orizzontale dell'asse dei valori; valori ammessi: <i>vertical</i> , <i>horizontal</i>
value	il valore per chart di tipo gauge
values	la serie di valori separati da virgola, per chart di tipo bars e points. E' possibile omettere alcuni valori della serie (es. values="1, 3, 2, , 4, , 5")
valuerange	valore minimo e massimo separati da virgola. Indicano il range del grafico. Es. "-100,100" (con orientation <i>vertical</i>) visualizzerà un grafico con valori da -100 a 100 sull'asse y
axislabels	etichette sull'asse x (se orientation è <i>vertical</i>) o sull'asse y (se orientation è <i>horizontal</i>). Non deve necessariamente corrispondere al numero di valori in values e può contenere label vuote (in tal caso vengono visualizzate solo le tacche)
drawaxis	specifica se gli assi (comprese le tacche e le label) vengono disegnati o meno; valori possibili: <i>true</i> o <i>false</i>
origin	sposta l'origine ad un valore diverso da 0 (ad esempio origin=3.2)
axisoffset	l'offset dell'asse (y se vertical, x se horizontal) in percentuale o come indice della serie (es. "50%" o "3")

attributo	valore
baroffset	sposta le barre o i punti del numero di pixel specificato, rendendo più semplice la sovrapposizione di più chart
notches	numero di tacche sull'asse x (se orientation è <i>horizontal</i>) o sull'asse y (se orientation è <i>vertical</i>)
spacing	spaziatura in percentuale fra le barre (valido solo per la modalità <i>bars</i>)
pointsize	dimensione dei "punti" (valido solo per la modalità <i>points</i>)
vlabelsstyle	stile delle value label; valori possibili: <i>inside</i> , <i>outside</i> , <i>none</i> . I valori non vengono visualizzati se questo attributo non è definito
bgcolor	colore dello sfondo in codice HTML
barcolor	serie di colori HTML, uno per ogni valore in values
barborder	visibilità del bordo delle barre/punti; valori possibili: <i>true</i> o <i>false</i>
axiscolor	colore HTML degli assi
notchcolor	colore HTML delle tacche sugli assi
labelcolor	colore HTML delle etichette sugli assi e sulle barre

(**datalogger <id>; <pos>; <width>; <height>; <label>; <attributes>**)

Questo oggetto è utilizzato per visualizzare i dati statistici elaborati dai data logger di HSYCO.



L'oggetto adatta automaticamente la posizione e le dimensioni dei grafici in funzione della dimensione dell'oggetto. La grafica di default può essere personalizzata utilizzando alcuni attributi opzionali.

Parametri:

- <id> - l'id del data logger, come specificato in hsyco.ini
- <pos> - posizione dell'oggetto. E' possibile utilizzare il formato coordinate X-Y in pixel o in righe-colonne
- <width> - larghezza dell'oggetto (chart) in pixel
- <height> - altezza dell'oggetto (chart) in pixel
- <label> - il nome della grandezza visualizzata
- <attributes> - attributi opzionali.

attribute	value
avgvalues	<i>true</i> : visualizza i valori medi; <i>false</i> : non visualizza i valori medi (solo per data logger di tipo range). Il valore di default è true
daily	<i>true</i> : visualizza i chart giornalieri; <i>false</i> : non visualizza i chart giornalieri. Il valore di default è true
hourly	<i>true</i> : visualizza i chart orari; <i>false</i> : non visualizza i chart orari. Il valore di default è true
legend	<i>false</i> : non visualizza la legenda dei valori; <i>true</i> : visualizza la legenda con i valori minimo, medio e massimo del periodo corrente e precedente. Impostare questo attributo a <i>min</i> , <i>avg</i> , <i>max</i> o qualunque combinazione di questi valori separati da virgola, per visualizzare la legenda con i valori dei rispettivi gruppi di dati. Il valore di default è true
maxvalues	<i>true</i> : visualizza i valori massimi; <i>false</i> : non visualizza i valori massimi (solo per data logger di tipo range). Il valore di default è true
minvalues	<i>true</i> : visualizza i valori minimi; <i>false</i> : non visualizza i valori minimi (solo per data logger di tipo range). Il valore di default è true
monthly	<i>true</i> : visualizza i chart mensili; <i>false</i> : non visualizza i chart mensili. Il valore di default è true
panel	<i>true</i> : visualizza il pannello di sfondo; <i>false</i> : non visualizza il pannello di sfondo. Il valore di default è true
pastbarscolor	colore delle barre dei valori del periodo precedente; utilizzare la rappresentazione standard HTML del colore. Specificare una lista di tre colori separati da virgola per i data logger di tipo range
pastperiod	<i>true</i> : visualizza i chart del periodo precedente; <i>false</i> : non visualizza i chart del periodo precedente. Il valore di default è true
presentbarscolor	colore delle barre dei valori del periodo corrente; utilizzare la rappresentazione standard HTML del colore. Specificare una lista di tre colori separati da virgola per i data logger di tipo range
tabs	<i>true</i> : visualizza separatamente i chart orari, giornalieri e mensili; <i>false</i> : visualizza i chart contemporaneamente. Il valore di default è true

attribute	value
totals	Se la legenda è visibile, impostare a true per visualizzare i valori del periodo corrente e precedente, oppure a false per visualizzare solo il testo min, med, max per i data logger di tipo range. Il valore di default è true
axiscolor	colore degli assi; utilizzare la rappresentazione standard HTML del colore
bgcolor	colore dello sfondo; utilizzare la rappresentazione standard HTML del colore
labelcolor	colore HTML delle etichette sugli assi e sulle barre
notches	numero di tacche sull'asse x
notchcolor	colore HTML delle tacche sugli assi
vlabelstyle	stile delle value label; valori possibili: <i>inside</i> , <i>outside</i> , <i>none</i> . I valori non vengono visualizzati se questo attributo non è definito

(video <src>; <pos>; <width>; <height>; <mode>)



Questo oggetto consente di visualizzare un video. Secondo lo standard HTML5 i formati video supportati sono: mp4, webm, ogv.

Per l'oggetto (video) esiste la versione identificata:

(video!id <src>; <pos>; <width>; <height>; <mode>)

Essa consente di controllare dinamicamente la posizione, la visibilità e la modalità di riproduzione.

Parametri:

<src> - nome del file senza estensione, presente nella directory www/img

<pos> - posizione del video all'interno della pagina. E' possibile utilizzare il formato coordinate X-Y in pixel o in righe-colonne

<width> - larghezza dell'oggetto (video) espressa in pixel

<height> - altezza dell'oggetto (video) espressa in pixel

<mode> - parametro opzionale. Può assumere i valori:

stop: valore di default. Il video non è in riproduzione

auto: quando la pagina è visualizzata riproduce automaticamente il video

loop: come auto, con riproduzione continua del video.

Amministrazione

(adminlink <function>; <pos>; <color>; <label>)

Questo oggetto crea il bottone di accesso alle pagine standard di configurazione. HSYCO prevede quattro pagine di configurazione: *users*, *clock*, *network* e *password*. Utilizzare il campo <function> dell'oggetto adminlink per accedere a ognuna di queste pagine.

Nella pagina *users* è possibile gestire i diritti d'accesso degli utenti. E' possibile impostare e revocare i diritti di amministratore, abilitare e disabilitare un utente, modificare PIN e PUK, eliminare un utente e definire i diritti di accesso alle sotto-directory.

La pagina *clock* imposta l'orologio del server e il fuso orario, e abilita l'aggiornamento automatico dell'ora basato sull'ora dei server da Internet.

La pagina *network* è usata per modificare i parametri IP delle porte Ethernet e della configurazione DNS. La pagina non è necessaria per il normale uso di HSYCO dopo la configurazione iniziale.

La pagina *password* è usata per cambiare la password di sistema. La password di sistema serve per accedere alla console del sistema operativo, direttamente o tramite SSH, e ai file di HSYCO attraverso il servizio di condivisione in rete dei file. Questa password è necessaria per accedere a HSYCO SERVER per attività di manutenzione e configurazione. Le pagine di amministrazione sono abilitate solamente per gli utenti con diritti di amministratore. A un utente senza diritti di amministratore che tenterà di accedere a queste pagine verrà mostrato un messaggio di errore.

Parametri:

<function> - funzione del bottone. Può assumere i valori: *users* - *clock* - *network* - *password*

<pos> - posizione del bottone all'interno della pagina. E' possibile utilizzare il formato coordinate X-Y in pixel o in righe-colonne

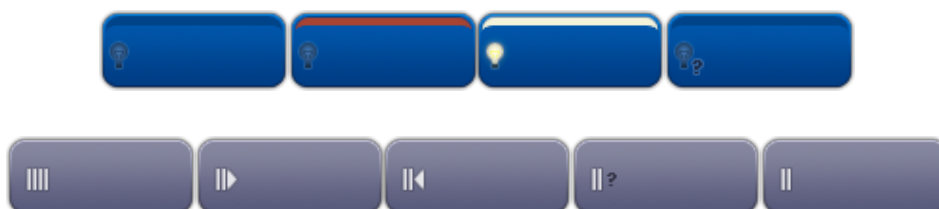
<color> - colore del bottone. Può assumere i valori: *b* - blu; *g* - verde; *gr* - grigio; *r* - rosso; *y* - giallo

<label> - testo presentato all'interno del bottone. Esso può contenere anche istruzioni HTML.

Luci e Automazioni

(button <address>; <pos>; <label>)

Controlla un dispositivo di illuminazione o di automazione.



Per i dispositivi di tipo VSHUT e HSHUT, associati a dispositivi di automazione in configurazione interblocco, premendo in sequenza questo bottone si ottengono le seguenti azioni:

UNKNOWN -> DOWN -> STOP -> UP -> STOP -> DOWN

Parametri:

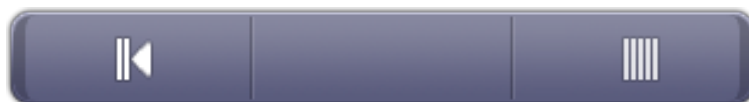
<address> - nome del data point del dispositivo

<pos> - posizione del bottone all'interno della pagina. E' possibile utilizzare il formato coordinate X-Y in pixel o in righe-colonne

<label> - testo presentato all'interno del bottone. Esso può contenere anche istruzioni HTML.

(3button <address>; <pos>; <label>)

(3button) è simile a (button). E' un unico bottone suddiviso in tre aree: salita/apertura, stop e discesa/chiusura.



Parametri:

<address> - nome del data point del dispositivo

<pos> - posizione del bottone all'interno della pagina. E' possibile utilizzare il formato coordinate X-Y in pixel o in righe-colonne

<label> - testo presentato all'interno del bottone. Esso può contenere anche istruzioni HTML.

(buttonicon <address>; <pos>; <color>)

(buttonicon) controlla un dispositivo di automazione o illuminazione, e ha la stessa funzione di (button).



Parametri:

<address> - nome del data point del dispositivo

<pos> - posizione del bottone all'interno della pagina. E' possibile utilizzare il formato coordinate X-Y in pixel o in righe-colonne

<color> - colore del bottone. Può assumere i valori:

gr - grigio pieno;

glass - bottone semi trasparente, che può essere sovrapposto a immagini di sfondo.

(buttonimage <address>; <pos>; <width>; <height>; <unknown img>; <on img>; <off img>; <up img>; <down img>; <offup img>; <offdown img>; <text>; <css>)

(buttonimage) controlla un dispositivo di automazione o illuminazione, e ha la stessa funzione di (button), ma è possibile scegliere immagini grafiche personalizzate per rappresentare i diversi stati. E' possibile visualizzare un testo opzionale associato all'immagine, che può essere formattato e posizionato grazie agli attributi standard CSS. In particolare, l'attributo "top" consente di riposizionare il testo. Se top non è definito, il testo sarà posizionato subito sotto l'immagine. Definendo un valore positivo o negativo in pixel è possibile spostare il testo rispettivamente in basso o in alto a partire dall'angolo in alto dell'immagine.

Parametri:

- <address> - nome del data point del dispositivo
- <pos> - posizione del bottone. E' possibile utilizzare il formato coordinate X-Y in pixel o in righe-colonne
- <width> - larghezza dell'oggetto, in pixel
- <height> - altezza dell'oggetto, in pixel
- <unknown img> - nome del file immagine associata allo stato sconosciuto
- <on img> - nome del file dell'immagine associata allo stato acceso
- <off img> - nome del file dell'immagine associata allo stato spento
- <up img> - nome del file dell'immagine associata allo stato su
- <down img> - nome del file dell'immagine associata allo stato giù
- <offup img> - nome del file dell'immagine associata allo stato fermo su
- <offdown img> - nome del file dell'immagine associata allo stato fermo giù.
- <text> - parametro opzionale. Testo, visualizzato normalmente sotto l'immagine
- <css> - parametro opzionale. Definisce lo stile CSS del testo.

Tutte le immagini vanno collocate nella directory `www/img` o in `www/<project>/img`

(dimmer <address>; <pos>; <label>)

Questo oggetto corrisponde a un attuatore di illuminazione di tipo dimmer.

Parametri:

- <address> - nome del data point del dispositivo
- <pos> - posizione del bottone all'interno della pagina. E' possibile utilizzare il formato coordinate X-Y in pixel o in righe-colonne
- <label> - testo presentato all'interno del bottone. Esso può contenere anche istruzioni HTML.

**(dmx <address>; <pos>; <label>)**

Questo oggetto consente il controllo di un singolo canale DMX.

Parametri:

- <address> - se è presente un solo gateway DMX, indirizzo sarà un numero compreso tra 1 e 512, come previsto dallo standard DMX-512. Se sono presenti più gateway, al numero di base deve essere sommato 1000 per il secondo gateway, 2000 per il terzo ecc. Ad esempio, 2100 indica l'indirizzo 100 sul bus DMX gestito dal terzo gateway
- <pos> - posizione del bottone all'interno della pagina. E' possibile utilizzare il formato coordinate X-Y in pixel o in righe-colonne.
- <label> - testo presentato all'interno del bottone. Esso può contenere anche istruzioni HTML.



(dmxrgb <address>; <pos>; <label>)

Questo oggetto consente il controllo di un gruppo di tre canali DMX per i colori fondamentali rosso, verde e blu.

Parametri:

<address> se è presente un solo gateway DMX, indirizzo sarà un numero compreso tra 1 e 512, come previsto dallo standard

DMX-512. Se sono presenti più gateway, al numero di base deve essere sommato 1000 per il secondo gateway, 2000 per il terzo ecc. Ad esempio, 2100 indica l'indirizzo 100 sul bus DMX gestito dal terzo gateway

<pos> - posizione del bottone all'interno della pagina. E' possibile utilizzare il formato coordinate X-Y in pixel o in righe-colonne.

<label> - testo presentato all'interno del bottone. Esso può contenere anche istruzioni HTML.



Controllo Temperatura

(temp <server>; <address>; <label>; <pos>)

Questo oggetto mostra un pannello per la visualizzazione dello stato e il controllo di una zona clima. Ha una grafica e funzioni differenti in base al sistema di controllo temperatura utilizzato. Consultare le Application Notes per ulteriori dettagli.



Parametri:

<server> - nome dell'IO Server

<address> - indirizzo della zona, vedi le Application Notes

<label> - testo descrittivo sul bottone. Esso può contenere anche istruzioni HTML

<pos> - posizione dell'oggetto all'interno della pagina. E' possibile utilizzare il formato coordinate X-Y in pixel o in righe-colonne.

(tempmini <server>; <address>; <label>; <pos>)

Questo oggetto mostra un pannello per la visualizzazione dello stato e il controllo di una zona clima. Ha una grafica e funzioni differenti in base al sistema di controllo temperatura utilizzato. Consultare le Application Notes per ulteriori dettagli.

**Parametri:**

<server> - nome dell'IO Server

<address> - indirizzo della zona, vedi le Application Notes

<label> - testo descrittivo sul bottone. Esso può contenere anche istruzioni HTML

<pos> - posizione dell'oggetto all'interno della pagina. E' possibile utilizzare il formato coordinate X-Y in pixel o in righe-colonne.

Timer e Scheduler**(timer <id>; <pos>; <label>)**

Questo oggetto corrisponde al mini-display per la visualizzazione dello stato ed il controllo di un timer. Il mini-display consente l'accesso alla pagina di gestione del timer, ma visualizza anche lo stato corrente del timer.

**Parametri:**

<id> - identificativo del timer, come definito nel parametro *Timers* nel file hsyco.ini

<pos> - posizione del bottone all'interno della pagina. E' possibile utilizzare il formato coordinate X-Y in pixel o in righe-colonne

<label> - testo visualizzato nel mini-display e nella zona centrale in basso del display quando viene visualizzata la pagina di controllo di ogni timer.

(scheduler <pos>; <width>; <height>; <mode>; <names>)

Questo oggetto fornisce l'interfaccia utente di uno scheduler.

Lo scheduler è una funzione che consente di definire un insieme di regole per programmare determinate azioni da eseguire sulla base dell'intervallo di tempo impostato.

La regola di base di uno scheduler è definita a partire da una data/tempo di inizio e fine ed è valida per tutto il periodo di tempo definito, anche per diversi giorni. Una regola giornaliera è invece ritenuta valida a partire da un tempo di inizio e uno di fine, di un singolo giorno o estesa a più giorni.

Oltre alle date di validità, è possibile impostare che una regola si attiva solo per specifici giorni della settimana.

Se l'ora corrente corrisponde esattamente al periodo di tempo definito nella regola dello scheduler, verranno generati l'evento TIMER <schedule name> = ON e il corrispondente metodo `userTimerEvent()` in Java - come l'oggetto (timer) - e l'evento TIMER <schedule name> = OFF alla fine del periodo di tempo.

E' anche possibile definire una regola di intervallo, impostando i minuti di on e di off. Una regola di intervallo sarà on e off in ripetutamente durante il periodo di tempo valido definito.

Ogni regola ha un nome. Tale nome viene utilizzato per generare gli eventi, non è univoco, ed è possibile definire diverse regole con lo stesso nome.

In questo caso lo scheduler valuterà le regole che hanno lo stesso nome nell'ordine con cui appaiono all'interno dello scheduler, dall'alto al basso.

Quando una regola trova corrispondenza con il periodo di tempo di validità definito, lo scheduler smetterà di cercare ogni altra regola successiva dichiarata

con lo stesso nome. In questo modo, le regole collocate in alto hanno priorità all'interno dello stesso gruppo di regole aventi lo stesso nome.

Le regole impostate per funzionare in modalità off vengono chiamate “regole bloccanti”, poichè quando sono valide non diventano attive, e allo stesso tempo, evitano che lo scheduler processi le regole seguenti che invece potrebbero corrispondere.

Parametri:

- <pos> - posizione dell'oggetto. E' possibile utilizzare il formato coordinate X-Y in pixel o in righe-colonne
- <width> - larghezza espressa in pixel
- <height> - altezza espressa in pixel
- <mode> - impostare a “edit” per abilitare la creazione o l'eliminazione di scheduler singoli. Impostare a “noedit” per consentire solamente la modifica di scheduler già definiti
- <names> - opzionale, lista di scheduler separati da virgole da mostrare all'interno dell'oggetto. Se non è definita alcuna lista o è impostato al valore “*”, l'oggetto mostra tutti gli scheduler definiti.

Telecamere

(cameralink <cam id>; <pos>; <color>; <label>; <dest panel id>)

Questo oggetto rappresenta un bottone, graficamente identico a (link), per l'accesso alla pagina di visualizzazione e gestione di una singola telecamera o di una griglia di telecamere.

Per funzionare correttamente, questo oggetto richiede la presenza della direttiva (#cameralist).

Parametri:

- <cam id> - nome della telecamera, come definito nel parametro Cameras in hsyco.ini, o il nome della griglia, definito come gridN, dove N è il numero della griglia assegnato in hsyco.ini
- <pos> - posizione dell'oggetto all'interno della pagina. E' possibile utilizzare il formato coordinate X-Y in pixel o in righe-colonne
- <color> - colore del bottone. Può assumere i valori: *b* - blu; *g* - verde; *gr* - grigio; *r* - rosso; *y* - giallo
- <label> - testo presentato all'interno del bottone. Esso può contenere anche istruzioni HTML
- <dest panel id> - Se impostato con l'id di un camera panel, cliccando sul bottone la telecamera sarà visualizzata sul camera panel specificato. Se impostato a "default", la telecamera sarà visualizzata su un camera panel dinamico alla massima dimensione.

(camera <cam id>; <pos>; <width>; <height>; <dest panel id>)



Questo oggetto visualizza l'immagine in tempo reale di una telecamera o di una matrice di telecamere.

Per l'oggetto (camera) esiste la versione identificata:

(camera!id <cam id>; <position>; <width>; <height>; <dest panel id>)

Parametri:

- <cam id> - nome della telecamera, come definito nel parametro Cameras in hsyco.ini, o il nome della griglia, definito come gridN, dove N è il numero della griglia assegnato in hsyco.ini
- <pos> - posizione dell'oggetto all'interno della pagina. E' possibile utilizzare il formato coordinate X-Y in pixel o in righe-colonne
- <width> - larghezza dell'immagine della telecamera, in pixel
- <height> - altezza dell'immagine della telecamera in pixel
- <dest panel id> - opzionale. Se impostato con l'id di un camera panel, cliccando sul bottone la telecamera sarà visualizzata sul camera panel specificato. Se impostato a "default", la telecamera sarà visualizzata su un camera panel dinamico alla massima dimensione.

(camerapanel <cam id>; <pos>; <width>; <height>; <cam list>)

!id

Questo oggetto mostra la visualizzazione real-time della telecamera o di una griglia di telecamere. A differenza dell'oggetto (camera), l'immagine è racchiusa all'interno di un pannello da cui è possibile gestire le funzioni PTZ e l'accesso alle immagini registrate.

Per l'oggetto (camerapanel) esiste la versione identificata:

(camerapanellid <cam id>; <position>; <width>; <height>; <cam list>)

Parametri:

- <cam id> - nome della telecamera, come definito nel parametro Cameras in hsyco.ini, o il nome della griglia, definito come gridN, dove N è il numero della griglia assegnato in hsyco.ini
- <pos> - posizione dell'oggetto. E' possibile utilizzare il formato coordinate X-Y in pixel o in righe-colonne
- <width> - larghezza del camera panel, in pixel
- <height> - altezza del camera panel in pixel
- <cam list> - opzionale. Se definito, limita la visualizzazione ciclica delle telecamere alle sole selezionate (cliccando sul nome della telecamera, o sull'immagine della telecamera stessa)

IRTrans

(ir <irid>; <com>; <pos>; <color>; <label>)

(irmini <irid>; <com>; <pos>; <color>; <label>)

(irmicro <irid>; <com>; <pos>; <color>; <label>)

Questo oggetto rappresenta un bottone per l'invio di comandi infrarossi (IR) attraverso gli IRTrans gestiti da HSYCO.

Parametri:

<irid> - identificativo dell'IRTrans e deve corrispondere ad uno dei nomi presenti nella lista del parametro IRTrans in hsyco.ini

<com> - sequenza di comandi IR da inviare. Ogni comando è costituito dal nome che identifica il database di comandi, seguito da virgola e dal nome del comando all'interno del database. Più comandi in sequenza possono essere separati da spazi. E' possibile inserire il comando speciale @N, che introduce un tempo di attesa pari a N millisecondi prima di inviare il comando successivo. Ad esempio:

panasonic-z40,onoff denon-1036,on @4000 denon-1036,tv

invia il comando onoff del telecomando panasonic-z40, seguito immediatamente dal comando on del telecomando denon-1036 ed infine dal comando tv sempre del denon-1036, inviando quest'ultimo comando dopo 4 secondi di pausa

<pos> - posizione del bottone all'interno della pagina. E' possibile utilizzare il formato coordinate X-Y in pixel o in righe-colonne

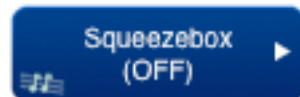
<color> - colore del bottone. Può assumere i valori: *b* - blu; *g* - verde; *gr* - grigio; *r* - rosso; *y* - giallo

<label> - testo presentato all'interno del bottone. Esso può contenere anche istruzioni HTML.

Player musicali

(music <id>; <pos>; <label>)

Questo oggetto serve ad inserire il mini-display per la gestione di uno Squeezebox all'interno della pagina.



Parametri:

- <id> - numero progressivo che identifica lo Squeezebox, a partire da 0, nella lista definita dal parametro slimPlayers in hsyco.ini
- <pos> - posizione dell'oggetto. E' possibile utilizzare il formato coordinate X-Y in pixel o in righe-colonne
- <label> - testo visualizzato sul bottone. Esso può contenere anche istruzioni HTML.

(musicsync <pos>)

Questo oggetto è usato per inserire il bottone [Sync] nella pagina standard di gestione della musica.

Parametri:

- <pos> - posizione del bottone all'interno della pagina. E' possibile utilizzare il formato coordinate X-Y in pixel o in righe-colonne.

(nuvo <id>; <zone>; <pos>)

(nuvomini <id>; <zone>; <pos>)

Questo oggetto è utilizzato per mostrare il display per la gestione del sistema multi-room NuVo Grand Concerto e Essentia.

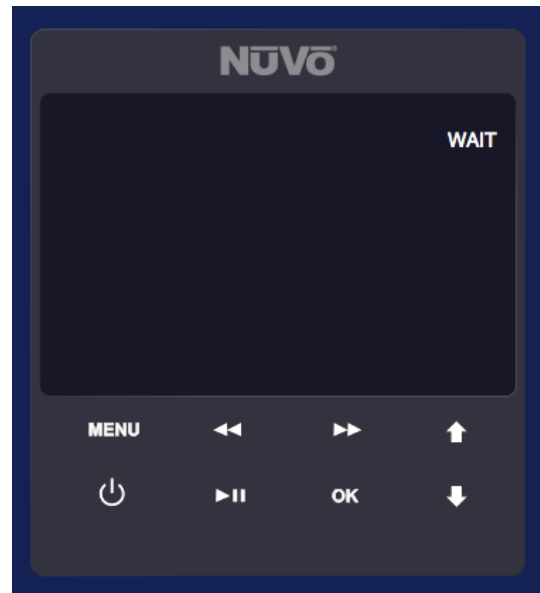
(nuvomini) offre le stesse funzioni e display di (nuvo), ma in un formato più piccolo, che si adatta meglio alle dimensioni standard di pagine per dispositivi mobili.

Parametri:

<id> - id dell'IO Server assegnato al sistema NuVo. Vedi Application Notes per ulteriori dettagli

<zone> - numero di zona

<pos> - posizione dell'oggetto. E' possibile utilizzare il formato coordinate X-Y in pixel o in righe-colonne.



Tasti USER

(user <name>; <param>; <pos>; <color>; <label>)

!id

(usermini <name>; <param>; <pos>; <color>; <label>)

!id

(usermicro <name>; <param>; <pos>; <color>; <label>)

!id

Questo oggetto consente di associare codice Java utente o azioni definite in events.txt ad un bottone.

Premendo il bottone creato con l'oggetto (user), HSYCO richiama il metodo:

userCommand(String name, String param)

nella classe *user.class*, passando i parametri *name* e *param* definiti nei corrispondenti campi dell'oggetto (user). HSYCO esegue inoltre le azioni associate all'evento:

USER name=param

definite in EVENTS. Questa funzione è descritta in dettaglio nel capitolo Programmazione.

Per gli oggetti (user), (usermini) e (usermicro) esiste la versione identificata:

(user!id <name>; <param>; <pos>; <color>; <label>)

(usermini!id <name>; <param>; <pos>; <color>; <label>)

(usermicro!id <name>; <param>; <pos>; <color>; <label>)



user



usermini



usermicro

Parametri:

<name> - parametro passato al metodo userCommand() o all'evento USER in EVENTS

<param> - parametro passato al metodo userCommand() o all'evento USER in EVENTS

<pos> - posizione dell'oggetto. E' possibile utilizzare il formato coordinate X-Y in pixel o in righe-colonne

<color> - colore del bottone. Può assumere i valori: *b* - blu; *g* - verde; *gr* - grigio; *r* - rosso; *y* - giallo

<label> - testo presentato sul bottone. Esso può contenere anche istruzioni HTML.

(userrgb <name>; <param>; <pos>; <color>; <label>)

!id

Questo oggetto consente di associare codice Java utente o azioni definite in EVENTS ad un bottone. E' analogo all'oggetto (user) ma consente di definire un colore generico, visualizzato nella parte centrale del bottone.

Per l'oggetto (userrgb) esiste la versione identificata:

(userrgb!id <name>; <param>; <pos>; <color>; <label>)

**Parametri:**

<name> - parametro passato al metodo userCommand() o all'evento USER in EVENTS

<param> - parametro passato al metodo userCommand() o all'evento USER in EVENTS

<pos> - posizione del bottone all'interno della pagina. E' possibile utilizzare il formato coordinate X-Y in pixel o in righe-colonne

<color> - colore del bottone. Esso deve essere specificato nel formato esadecimale rrggbb, dove rr, gg e bb sono i valori esadecimali da 00 a FF dei colori rosso, verde e blu. Ad esempio:

FF0000 - rosso

00FF00 - verde

0000FF - blu

000000 - nero

FFFFFF - bianco

FFFF00 - giallo

<label> - testo presentato sul bottone. Esso può contenere anche istruzioni HTML.

**(userimage <file>;<pos>;<width>;<height>;<name>;<param>; <text>;
<css>)** 

Questo oggetto consente di associare codice Java utente o azioni definite in events.txt ad un bottone. E' analogo all'oggetto (user) ma consente di definire un'immagine generica al posto di quella standard. E' possibile visualizzare un testo opzionale associato all'immagine, che può essere formattato e posizionato grazie agli attributi standard CSS. In particolare, l'attributo "top" consente di riposizionare il testo. Se top non è definito, il testo sarà posizionato subito sotto l'immagine. Definendo un valore positivo o negativo in pixel è possibile spostare il testo rispettivamente in basso o in alto a partire dall'angolo in alto dell'immagine.

Per l'oggetto (userimage) esiste la versione identificata:

(userimage!id <file>; <pos>; <width>; <height>; <name>; <param>;
<text>; <css>)

Parametri:

- <file> - nome del file da utilizzare come immagine, presente nella directory
www/img
- <pos> - posizione dell'oggetto. E' possibile utilizzare il formato coordinate
X-Y in pixel o in righe-colonne
- <width> - larghezza dell'immagine. Essa deve essere specificata in pixel
- <height> - altezza dell'immagine. Essa deve essere specificata in pixel
- <name> - parametro passato al metodo userCommand() o all'evento
USER in EVENTS
- <param> - parametro passato al metodo userCommand() o all'evento
USER in EVENTS
- <text> - parametro opzionale. Testo, visualizzato normalmente sotto
l'immagine
- <css> - parametro opzionale. Definisce lo stile CSS del testo.

(slider!id <name>; <pos>; <label>)

!id

Questo oggetto consente di associare codice Java utente o azioni definite in events.txt ad un bottone, passando il nome dell'oggetto e il valore selezionato. E' analogo all'oggetto (user) ma consente di definire uno slider con una barra di regolazione.

Per l'oggetto (slider) esiste solo la versione identificata.

Parametri:

- <name> - parametro passato al metodo userCommand() o
all'evento USER in EVENTS
- <pos> - posizione dell'oggetto. E' possibile utilizzare il
formato coordinate X-Y in pixel o in righe-colonne
- <label> - testo presentato all'interno del bottone. Esso può contenere
anche istruzioni HTML.



(sliderbutton!id <name>; <pos>; <label>)

Analogamente a (slider), questo oggetto consente di associare codice Java utente o azioni definite in events.txt ad un bottone, passando il nome dell'oggetto e il valore selezionato.

Per l'oggetto (sliderbutton) esiste solo la versione identificata.

Parametri:

<name> - parametro passato al metodo
userCommand() o all'evento USER in
EVENTS

<pos> - posizione dell'oggetto. E' possibile
utilizzare il formato coordinate X-Y in pixel o in righe-colonne

<label> - testo presentato all'interno del bottone. Esso può contenere
anche istruzioni HTML.



Form

(input!id <pos>; <css>)

!id

Questo oggetto visualizza un campo di testo in un form.

Per l'oggetto (input) esiste solo la versione identificata.

Parametri:

<pos> - posizione della casella di testo all'interno della pagina. E' possibile utilizzare il formato coordinate X-Y in pixel o in righe-colonne

<css> - parametro opzionale. Definisce lo stile CSS del campo.

(submit!id <pos>; <color>; <label>)

!id

(submitmini!id <pos>; <color>; <label>)

!id

(submitmicro!id <pos>; <color>; <label>)

!id

Questo oggetto consente di associare ad un bottone codice Java personalizzato. Premendo il bottone creato con l'oggetto (submit), HSYCO richiama il metodo:

userCommand(String name, String param)

nella classe *user.class*, passando come parametri:

name - l'id dell'oggetto (submit)

param - una stringa composta da tutti gli id degli oggetti (input) e il testo.

Per gli oggetti (submit), (submitmini), (submitmicro) esiste solo la versione identificata.



Se l'id dell'oggetto (submit!id) inizia con il carattere \$, allora HSYCO imposterà automaticamente variabili con nome \$<id>.<input id> ai valori di ciascun campo di input nel form.

Parametri:

- <pos> - posizione dell'oggetto. E' possibile utilizzare il formato coordinate X-Y in pixel o in righe-colonne
- <color> - colore del bottone. Può assumere i valori: *b* - blu; *g* - verde; *gr* - grigio; *r* - rosso; *y* - giallo
- <text> - testo visualizzato sul bottone.

(submitimage <file>; <pos>; <width>; <height>; <text>; <css>)

!id

Come (submit), questo oggetto viene utilizzato per eseguire codice Java personalizzato, passando il valore di tutti i campi input all'interno della stessa pagina, pop-up o container. Come l'oggetto (image), è rappresentato graficamente da un'immagine personalizzata e una stringa di testo.

Per l'oggetto (submitimage) esiste la versione identificata:

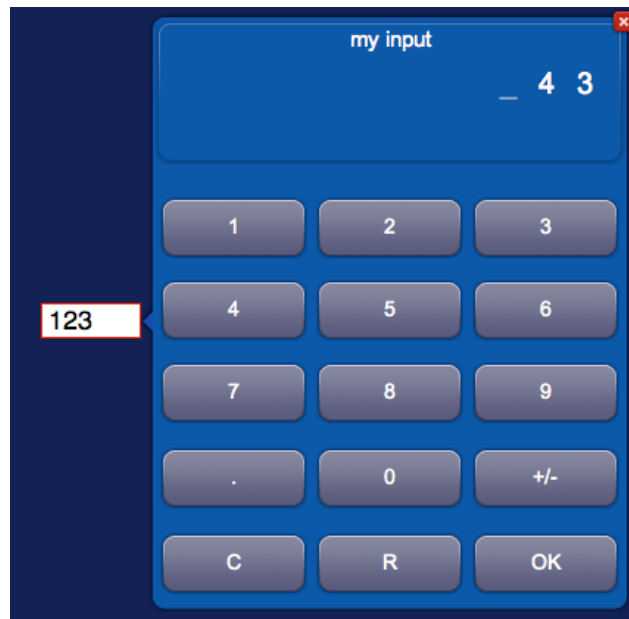
(submitimage <file>; <pos>; <width>; <height>; <text>; <css>)

Parametri:

- <file> - nome del file da utilizzare come immagine
- <pos> - posizione dell'oggetto. E' possibile utilizzare il formato coordinate X-Y in pixel o in righe-colonne
- <width> larghezza dell'immagine espressa in pixel
- <height> altezza dell'immagine espressa in pixel
- <text> - stringa di testo opzionale
- <css> - stile CSS opzionale.

**(keypad!id <pos>; <min>; <max>; <digits>; <decimals>; <type>; <label>;
<css>)** !id

Visualizza un campo di input numerico. Toccando il campo viene visualizzata una tastiera virtuale che consente di digitare i numeri. E' anche possibile inserire i numeri utilizzando la tastiera fisica.



Premendo il tasto OK, oppure Enter sulla tastiera fisica, vengono eseguiti i controlli sul valore e sul numero di cifre del numero inserito. In caso di errore viene visualizzato un messaggio, altrimenti HSYCO eseguirà il metodo:

userCommand(String name, String param)

della classe *user.class*, passando:

name - l'id dell'oggetto (submit)

param - il valore inserito.

Inoltre verrà generato l'evento USER <id>.

Se l'id dell'oggetto (keypad!id) inizia con il carattere \$, allora HSYCO imposterà automaticamente la variabile con nome \$<id> al valore inserito. L'oggetto (keypad) può anche essere usato nei form, come gli oggetti (input). L'identificativo dell'oggetto (keypad) è obbligatorio.

Parametri:

- <pos> - posizione dell'oggetto. E' possibile utilizzare il formato coordinate X-Y in pixel o in righe-colonne
- <min> - minimo valore ammesso
- <max> - massimo valore ammesso
- <digits> - numero di cifre obbligatorie (compresi eventuali decimali); lasciare vuoto per non avere vincoli sul numero di cifre inserite
- <decimals> - numero massimo di decimali; lasciare vuoto per numeri interi
- <type> - impostare a "input" per campi numerici normali; impostare a "password" per nascondere le cifre impostate
- <label> - testo libero visualizzato nella parte alta del keypad
- <css> - opzionale. Definisce lo stile CSS del campo.

Weather

(weather!id <pos>)

!id

Questo oggetto visualizza le condizioni meteo attuali e le previsioni per i due giorni successivi. Le informazioni meteo sono fornite dall'I/O Server WXONLINE.



L'id dell'oggetto deve corrispondere al nome dell'I/O Server dichiarato all'interno del file hsyco.ini. Per maggiori dettagli consultare l'Application Note dell'I/O Server WXONLINE.

Per l'oggetto (weather) esiste solo la versione identificata.

Parametri:

- <pos> - posizione dell'oggetto all'interno della pagina. E' possibile utilizzare il formato coordinate X-Y in pixel o in righe-colonne.

Le Opzioni di Visualizzazione

Lo skin standard “blue” prevede la possibilità di definire alcune opzioni in grado di alterare il comportamento dell’interfaccia Web.

Queste opzioni vengono attivate specificando delle parole chiave nell’url della pagina, sono quindi liberamente gestibili da ciascun Web Browser.

Il formato è il seguente:

https://<name server>/<name sub-directory>/pin.<urlkey>?<options>

Il campo opzioni può assumere uno o più dei valori descritti di seguito; parole chiave multiple possono essere separate dal carattere “.”:

- ***nocamstream***: disabilita lo streaming MJPEG del browser Web, anche se il browser supporta questo formato
- ***nocache***: disabilita la cache persistente HTML5 per la sessione Web, sovrascrivendo le impostazioni di default
- ***nopics***: disabilita la visualizzazione delle immagini associate ai dispositivi, come descritto per il file systemtopo.txt. Questa opzione viene normalmente utilizzata per migliorare le prestazioni nel caso di connessioni remote non sufficientemente veloci
- ***page=<page name>***: al caricamento iniziale dell’interfaccia nel browser Web, visualizza la pagina specificata invece del menu.

Lingua delle Pagine Web

La creazione di pagine di interfaccia Web in diverse lingue è molto semplice. Tutti i testi specifici per l'impianto si trovano direttamente nel file `index.hsm`, è quindi necessario creare un file `index.hsm` per ogni lingua, ricordando che ogni file deve trovarsi in una sotto-directory `www` dedicata.

La direttiva (`#language <id lingua>`) determina la selezione della lingua dei messaggi di testo relativi all'interfaccia Web. Lo skin standard di HSYCO prevede, oltre alla lingua italiana, anche messaggi in lingua inglese e francese, i cui id sono rispettivamente *en* e *fr*. Per aggiungere messaggi in un'altra lingua all'interno dello skin standard è sufficiente creare un file **`text_<id lingua>.txt`** nella directory dello skin.

Il file `text_<id lingua>.txt` contiene tutti i messaggi visualizzati o generati da HSYCO a livello di interfaccia Web.

Il formato di `text_<id lingua>.txt` è semplicemente:

nome = valore

Gli spazi a inizio e fine di nome e valore vengono ignorati, così come le righe vuote e le righe che iniziano con il carattere "#", che possono essere utilizzate per commenti.

E' opportuno tradurre tutte le righe presenti in `text_<id lingua>.txt`, anche eventuali testi non necessari per una specifica applicazione, per evitare possibili malfunzionamenti a livello di interfaccia Web.

Alcuni esempi:

```
weekdayslist = "L","M","M","G","V","S","D"
pressctocont = Premere C per continuare
enterpuk = Inserire il PUK e premere OK

weekdayslist = "M","T","W","T","F","S","S"
pressctocont = Press C to continue
enterpuk = Enter PUK code then press OK
```


Il Funzionamento dell'Interprete di Skin

Il formato del file index.hsm descritto in questo capitolo e le funzionalità elencate non sono rigidamente imposte da HSYCO. In realtà tutto quanto è stato scritto relativamente alla costruzione delle pagine dell'interfaccia Web può essere completamente modificato.

Infatti gli oggetti fino ad ora descritti non sono altro che semplici macro definite nello skin standard, che possono essere facilmente modificate, estese con nuovi oggetti o sostituite completamente con oggetti diversi.

Per capire come questo sia possibile è sufficiente descrivere il funzionamento del sistema, integrato nel Web Server di HSYCO, che si occupa della traduzione di index.hsm in codice HTML leggibile da qualunque Web Browser.

Il formato del file index.hsm è basato su oggetti, la cui rappresentazione generica è:

(tipo_oggetto <parametro1>;<parametroN>)

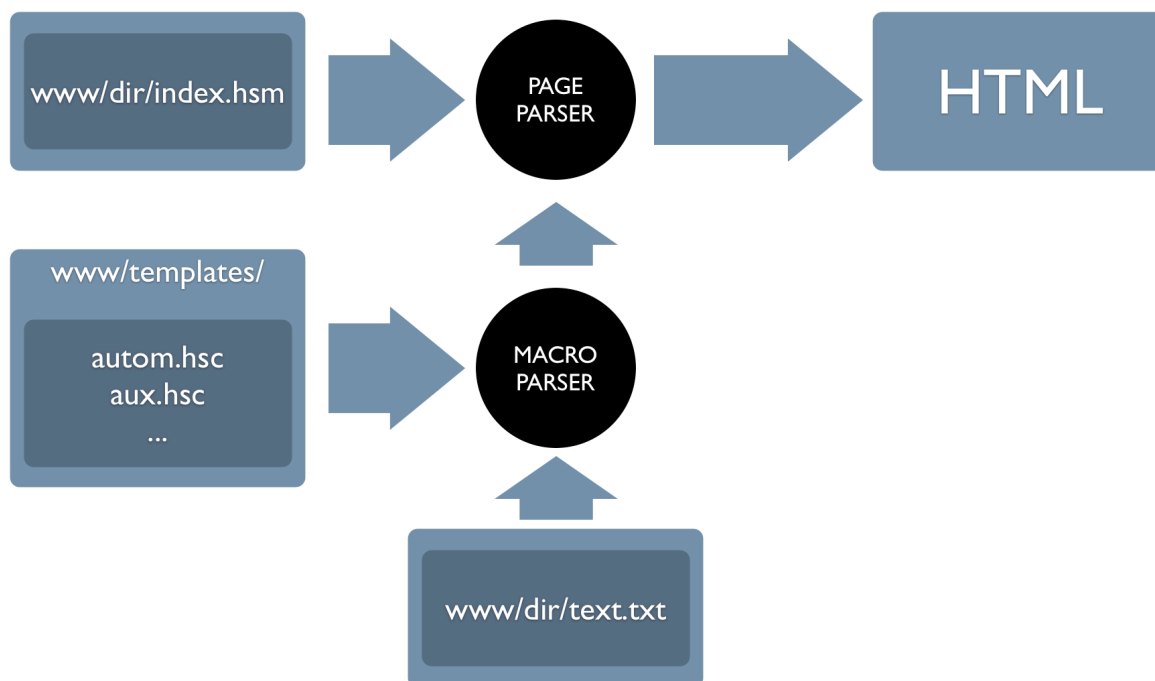
Ogni tipo di oggetto può richiedere o meno uno o più parametri, separati dal carattere “;”. Ad ogni richiesta ricevuta dal Web Server, il file viene letto dal **Page Parser**, che traduce il file come illustrato nello schema seguente.

Per ogni oggetto individuato nel file index.hsm, il Page Parser richiede al **Macro Parser** la traduzione dell'oggetto stesso con i parametri specificati.

Il lavoro svolto dal Macro Parser è molto semplice, consiste infatti nel leggere dalla directory `www/<nome skin>` il contenuto del file **tipo_oggetto.hsc**¹³.

Lo skin utilizzato per un file index.hsm è definito dalla direttiva **(#skin <nome skin>)** all'inizio del file, prima dell'oggetto (header).

¹³ Come già detto più volte, HSYCO cerca sempre i file prima nelle sotto-directory di `www` e, solo se il file non è presente, cerca all'interno di `hsyco.jar`. Quindi un file predefinito in HSYCO può sempre essere sostituito con un nuovo file.



Il Macro Parser si limita infatti a sostituire tutte le stringhe tipo **\$N** (dove N è un numero compreso tra 1 e 9¹⁴) con il corrispondente parametro presente nel corpo dell'oggetto definito in index.hsm. Sostituisce inoltre tutte le stringhe tipo **\$nome;** con il valore associato a *nome* nel file **text_<id lingua>.txt** che si trova nella directory dello skin utilizzato.

Per poter scrivere il carattere "\$" in quanto tale all'interno di file con estensione hsc, è necessario scrivere **\$\$**. Un carattere "\$" isolato costituisce un errore.

Allo stesso modo tutti gli altri file contengono frammenti più o meno grandi di codice HTML. Il risultato finale è un codice HTML completo, costruito a partire da tanti oggetti che nell'insieme descrivono l'interfaccia Web.

La maggior parte di questi file sono costituiti da una sola riga. Altri sono invece molto più grandi, come ad esempio **header.hsc**, che contiene tutta la definizione CSS di stile, quindi formati dei caratteri, dimensioni degli oggetti grafici e informazioni di posizione. Un altro file importante è **main.hsc**, che contiene il codice HTML delle pagine standard, dell'area display, e alcune righe di codice JavaScript necessario per la gestione multi-lingua dei testi e altre

¹⁴ Non è possibile definire oggetti con più di 9 parametri.

funzioni, nonché il caricamento del file JavaScript principale contenente l'applicazione Web vera e propria.

Infine il file **pin.hsc** contiene il codice HTML per la gestione dell'interfaccia Web di login. Quando il Web Server riceve una richiesta di accesso all'interfaccia Web, se il Web Browser non è già autenticato, viene restituito il contenuto di pin.hsc, non la pagina completa di gestione dell'interfaccia Web. Solo a seguito di una corretta autenticazione viene caricata la pagina completa corrispondente a index.hsm.

Tutti questi file sono modificabili, rendendo possibile un livello di personalizzazione pressoché totale dell'interfaccia Web di HSYCO.

Inserire Codice HTML nelle Pagine

In alcuni casi potrebbe essere opportuno scrivere direttamente codice HTML, invece di utilizzare la definizione a oggetti in index.hsm.

Tuttavia, poiché il Web Server di HSYCO non è progettato per servire file HTML puri ed il contenuto di index.hsm deve essere rigorosamente nel formato a oggetti, il modo raccomandato per inserire codice HTML è di creare file con estensione hsc all'interno della directory dello skin.

Ad esempio il contenuto HTML¹⁵ di un file `www/templates/custom/myHtml.hsc` può essere inserito in un punto appropriato di index.hsm semplicemente scrivendo:

```
(custom/myHtml)
```

L'unica accortezza è di seguire regole di nomenclatura degli oggetti che siano chiare e consentano una semplice gestione dei file associati agli oggetti, ricordando la precedenza dei file nelle directory rispetto a quelli standard presenti in hsyco.jar.

¹⁵ E' opportuno ricordare ancora una volta che i file hsc non possono contenere il carattere "\$" isolato, che deve essere scritto come \$\$\$. Inoltre nel contenuto di index.hsm e per i nomi delle variabili nei file hsc, le lettere maiuscole e minuscole sono significative, ad esempio (menu) e (MENU) sono due oggetti diversi, di cui il secondo non esiste nello skin standard.

Programmazione a Eventi

Le funzionalità base di HSYCO possono essere estese grazie a due sistemi di programmazione: la programmazione in linguaggio Java e la programmazione a eventi. La programmazione in linguaggio Java mette a disposizione una libreria di metodi per l'esecuzione di comandi verso tutti i sistemi controllati e offre la possibilità di associare codice Java personalizzato a qualunque evento registrato dal server HSYCO. EVENTS è un metodo complementare che, sebbene più veloce e più semplice rispetto al primo, permette comunque una gestione delle funzionalità efficace utilizzando una semplice sintassi all'interno di un file di testo.

L'architettura di HSYCO è estesamente *multi-threaded*¹⁶, per garantire per quanto possibile tempi di risposta rapidi rispetto a qualunque evento gestito. Ogni chiamata alle funzioni Java o agli eventi è realizzata in un **thread** indipendente, quindi non bloccante per le altre chiamate. Tuttavia, essendo le risorse hardware comunque condivise e limitate, elaborazioni particolarmente pesanti potrebbero avere un impatto negativo in termini di prestazioni e stabilità sull'intero sistema.

HSYCO, per svolgere le normali attività di supervisione e controllo, effettua un costante monitoraggio dello stato di tutti i sistemi di campo. In seguito a qualunque variazione di stato, come ad esempio l'accensione di un punto luce o la variazione della temperatura in una zona, ha la possibilità di eseguire chiamate a codice Java personalizzato o a EVENTS, rendendo così possibile l'implementazione di qualunque logica di controllo complessa, ma anche l'integrazione con altri sistemi.

¹⁶ In pratica le diverse attività del sistema vengono eseguite da diversi rami di esecuzione paralleli. In questo modo, anche se un ramo di esecuzione richiede un tempo particolarmente lungo per essere completato, questo non pregiudica necessariamente i tempi di risposta degli altri rami in esecuzione.

EVENTS

La programmazione a eventi è basata su un linguaggio molto semplice, che consente di associare una o più azioni ad un evento o ad una combinazione di eventi.

La directory principale di HSYCO contiene, oltre agli altri file di configurazione, il file di testo **events.txt**. Questo file permette la gestione della programmazione di HSYCO in maniera complementare e semplificata rispetto alla programmazione in linguaggio Java. Non è necessario eseguire alcun processo di compilazione; le modifiche vengono ricaricate automaticamente e diventano effettive non appena il file viene salvato.

Un evento è una espressione che rappresenta lo stato persistente o transitorio di uno o più dispositivi, condizioni applicate a variabili o un evento istantaneo. Un'azione è una sequenza di comandi verso un dispositivo o altre funzioni interne, come le impostazioni di variabili e di program timer.

Esempio:

```
IO k.33 = 1 : k.34 = 1
```

quando l'attuatore di illuminazione con data point 33 viene acceso, HSYCO accende automaticamente l'attuatore con data point 34.

Esempio:

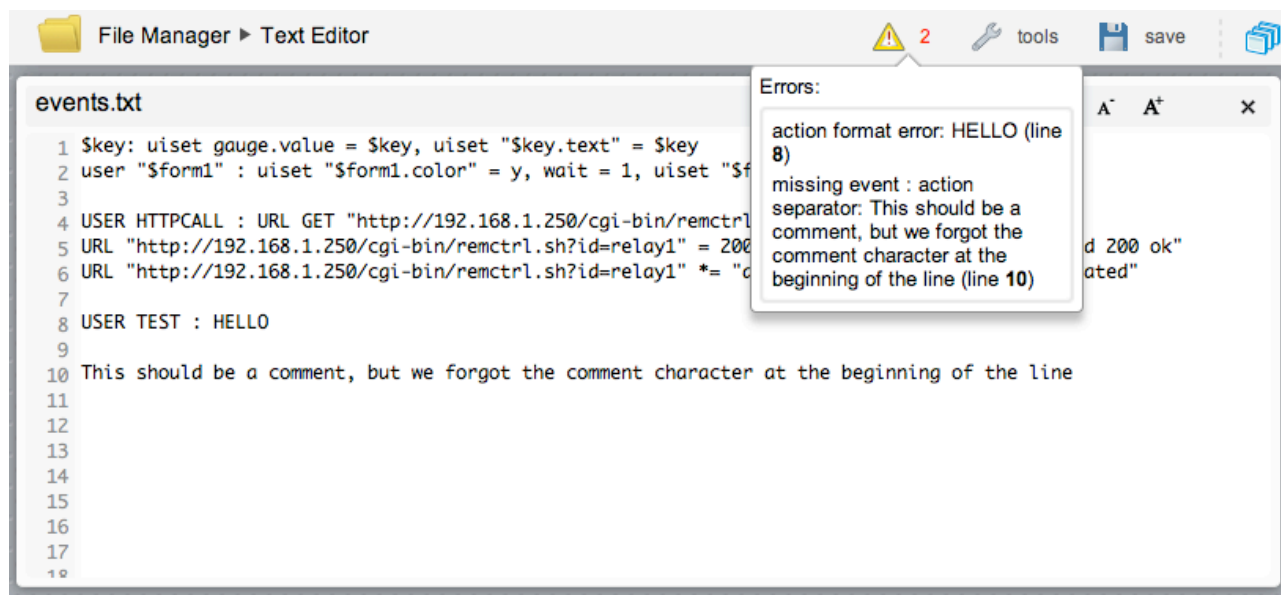
```
DAY : IO k.33 k.34 k.35 k.36 = 0
```

all'alba vengono spenti gli attuatori 33, 34, 35 e 36.

Nel seguito di questo capitolo viene descritto dettagliatamente il formato degli eventi e delle azioni disponibili per la personalizzazione di events.txt.

Il file events.txt viene normalmente modificato attraverso il text editor del File Manager di HSYCO. Il file viene verificato automaticamente nel momento in cui le modifiche sono salvate e, in caso di errori, viene visualizzata una icona di allerta nella barra in alto.

Toccando l'icona è possibile visualizzare l'elenco degli errori. Tutti gli errori vengono anche scritti nei file di log. Le righe di events.txt contenenti errori vengono ignorate.



Il linguaggio EVENTS

EVENTS utilizza il semplice formato

evento : azione

E' possibile combinare più eventi e realizzare espressioni complesse che rappresentano la combinazione di diverse condizioni.

Sono disponibili gli operatori logici AND, OR, NOT e le parentesi tonde per la priorità delle operazioni.

Se non vengono usate le parentesi, le operazioni AND e OR vengono eseguite da sinistra a destra, mentre l'operatore NOT ha la priorità e viene eseguito sull'espressione immediatamente alla sua destra, prima di AND e OR.

Esempio:

IO k.33 OR IO K.34

è un evento che si verifica ad un qualunque cambiamento di stato dell'attuatore 33 o dell'attuatore 34.

Esempio:

```
CAMERA ingresso AND NOT SECURITY = ALARM
```

è un evento che si verifica quando la telecamera “ingresso” inizia a registrare, ma solo se l’antifurto non è in stato di allarme.

Nota:

```
IO k.33 > 5 OR IO K.34 = 0 AND IO K.35 = 0
```

corrisponde a

```
(IO k.33 > 5 OR IO K.34 = 0) AND IO K.35 = 0
```

Nota:

```
NOT IO k.33 = 1 AND IO k.34 = 1
```

corrisponde a

```
(NOT IO k.33 = 1) AND IO k.34 = 1
```

Commenti

I commenti sono molto importanti per descrivere le funzioni svolte, anche se il linguaggio EVENTS è molto facile da comprendere.

Un commento è identificato dal carattere #.

Tutti i caratteri che seguono #, fino a fine riga vengono ignorati.

Esempio:

```
# azioni eseguite al tramonto
```

```
NIGHT:
```

```
IO k.33 k.34 k.35 k.36 = 1, # accende luci giardino
```

```
IO k.43 k.44 = UP # alza le tapparelle
```

Altre Regole Generali

Tutte le parole chiave possono essere scritte indifferentemente in maiuscolo o minuscolo. Anche i nomi dei dispositivi possono essere scritti in maiuscolo o minuscolo. Poiché l’interprete di EVENTS converte tutti gli id in minuscolo, è consigliato scrivere gli id solo in minuscolo nel file hsyco.ini.

I valori assegnati nelle azioni possono essere indifferentemente in maiuscolo o minuscolo per i valori standard *ON*, *OFF*, *UP*, *DOWN*, *STOP*, *FLIP*, *RECON*, *RECOFF*, *MERGE*, *UNMERGE*, *PLAY*. Negli altri casi invece i caratteri vengono mantenuti come scritti.

Esempio:

```
IO k.43 = UP and IO k.44 = DOWN : CAMERA cam1 = 60
```

è equivalente a:

```
io k.43 = up AND io k.44 = down : camera cam1 = 60
```

Esempio:

```
IO k.36 = 1 : UISET message.text = "Luce scale ON"
```

Il testo "Luce Scale ON" viene mantenuto come scritto.

Gli spazi e i caratteri di tabulazione non significativi vengono ignorati.

E' possibile non inserire gli spazi di separazione in alcuni casi:

```
IO k.36=1
```

è una espressione valida, invece

```
IOk.36 = 1
```

non è valida perché le parole chiave devono essere separate dagli attributi.

E' possibile usare i doppi apici per le stringhe, in particolare quando le stringhe contengono caratteri speciali, come , : oppure #

Esempio:

```
IO k.36 = 1 : LOG = "### Accendo la luce: 36"
```

Eventi Stabili e Transitori

```
IO k.33
```

è un evento *transitorio*. Si verifica solo al cambiamento di stato dell'attuatore k. 33 e, se inserito in una espressione composta, risulta vero solo quando

l'espressione viene valutata in corrispondenza dell'evento stesso, ma falso quando l'espressione viene valutata in corrispondenza di qualunque altro evento.

```
IO k.33 AND IO k.33
```

è un evento che non può mai verificarsi, perché due eventi transitori non possono mai verificarsi contemporaneamente.

```
IO k.33 > 40
```

è un evento *stabile*. Si verifica quando lo stato dell'attuatore k.33 assume un valore maggiore di 40 (più del 40% di livello di un dimmer).

Diversamente dagli eventi transitori, questa espressione è comunque vera anche dopo l'evento di transizione, quindi:

```
IO k.33 > 40 AND IO k.34 > 40
```

è un evento che può verificarsi, quando entrambi gli attuatori k.33 e k.34 sono accesi con livello superiore al 40%.

In alcuni casi può essere necessario utilizzare un evento transitorio associato ad un dispositivo, in abbinamento ad espressioni di stato anche dello stesso dispositivo.

Esempio:

```
IO k.33 = 1 AND IO k.34 = 1 AND NOT IO k.33
```

L'espressione NOT IO k.33 esclude gli eventi generati da qualunque variazione del dispositivo 33, ma allo stesso tempo k.33 = ON fa sì che l'espressione risulti vera solo quando il dispositivo è in stato ON.

Ne risulta che l'evento si verifica quando i dispositivi k.33 e k.34 sono contemporaneamente ON, ma solo sul cambiamento di stato del k.34, ignorando i cambiamenti di stato del k.33.

Esempio:

```
IO k.33 = 1 AND IO k.34 = 1 AND NOT IO k.33 :
```

```
IO k.33 = 0, WAIT = 0.5, IO k.33 = 1
```


Se la luce k.33 è già ON, nel momento in cui la luce k.34 viene accesa, la luce k.33 viene fatta lampeggiare, spegnendola e riaccendendola dopo una breve pausa.

Omettendo l'espressione NOT IO k.33 il lampeggio si ripeterebbe all'infinito, in quanto l'azione scatenerrebbe nuovamente lo stesso evento.

Un Evento, più Regole

E' possibile specificare più regole in cui sia presente lo stesso evento elementare.

Quando HSYCO rileva una qualunque variazione di stato, viene eseguita in EVENTS una ricerca di tutte le espressioni di evento interessate dalla variazione. Tutte le espressioni rilevanti vengono rivalutate e, se il risultato logico è vero, vengono eseguite le azioni associate. In genere l'esecuzione avviene in ordine dalla prima riga del file alle successive, ma quest'ordine di esecuzione non è garantito.

Esempio:

```
IO k.34 = 1 : IO k.36 = 0  
IO k.34 = 1 AND DAY : IO k.37 = 0
```

Queste due regole fanno sì che, quando l'attuatore k.34 viene acceso, l'attuatore k.36 viene spento in ogni caso, mentre l'attuatore k.37 viene spento solo durante il giorno.

Azioni

Ad ogni evento è possibile associare una singola azione, oppure una sequenza di azioni separate da virgola.

```
NIGHT : IO k.33 = 1, IO k.34 = 1, IO k.88 = UP
```

Per velocizzare la scrittura e rendere più leggibili le regole, molti comandi consentono anche di elencare una lista di indirizzi come attributo dello stesso comando. La regola scritta sopra può essere riscritta in modo più sintetico:

```
NIGHT : IO k.33 k.34 = 1, IO k.88 = UP
```

Normalmente ogni regola *evento : azione* è scritta su una stessa riga del file. Ma per migliorare la leggibilità è possibile continuare una lista di azioni sulla riga successiva, terminando la riga precedente con la virgola. E' anche possibile lasciare in una riga la sola espressione dell'evento seguita da : e iniziare la lista di azioni alla riga successiva:

```
NIGHT :
```

```
IO k.33 k.34 = 1, IO k.86 k.87 k.88 = UP
```

Non è consentito spezzare l'espressione dell'evento su più righe.

Variabili

Le variabili sono identificate da un nome qualunque, composto da lettere e numeri, che inizia con il carattere \$.

```
$livello
```

I caratteri che compongono il nome possono essere scritti indifferentemente in lettere maiuscole o minuscole. \$livello e \$LIVELLO indicano la stessa variabile.

Una variabile può contenere qualunque valore assegnato e non deve essere dichiarata prima di essere utilizzata.

Variabili e Azioni

E' possibile assegnare valori costanti alle variabili, con espressioni di assegnazione nelle azioni:

```
HSYCOSTART : $contatore = 0
```

assegna il valore 0 alla variabile \$contatore, nel momento in cui HSYCO viene avviato.

E' anche possibile eseguire le operazioni aritmetiche di base: +, -, * e /, il calcolo del resto % della divisione e l'arrotondamento (ROUND n) con un numero arbitrario¹⁷ di decimali. E' possibile concatenare il testo presente in una variabile con un altro testo, usando l'operatore +, che assume automaticamente la funzione di concatenazione se il contenuto originale della variabile, o l'operando non sono numerici.

Esempio:

```
IO k.33 = 1 : $count + 1
```

incrementa il valore della variabile \$contatore ogni volta che l'attuatore k.33 viene acceso.

Esempio:

```
IO modbus.27.240 : $energy = IO modbus.27.240,  
$energy ROUND 2, LOG $energy
```

alla variazione di un registro di un dispositivo MODBUS, imposta la variabile \$energy al valore del registro, arrotondando il valore a 2 decimali.

Esempio:

```
IO k.33 = 1 : $count + 1, $count % 4
```

incrementa come nell'esempio precedente, ma esegue poi la divisione per 4 del contenuto della variabile, scrivendo nella variabile stessa il resto della divisione. In questo modo il valore di questa variabile verrà incrementato fino a 3, per poi tornare al valore 0.

¹⁷ E' possibile effettuare l'arrotondamento da 0 a 9 decimali. L'arrotondamento a 0 decimali corrisponde all'arrotondamento al numero intero.

Per cancellare una variabile, assegnare ad essa una stringa vuota:

```
IO k.33 = 0 : $status = ""
```

E' possibile scrivere più operandi dopo l'uguale o in generale dopo l'operatore. In questo caso gli operandi saranno concatenati.

Questo è particolarmente comodo per assegnare ad una variabile un testo complesso.

Esempio:

```
time : $body = "Il consumo giornaliero " $date:y/m/d$  
" alle ore " $time:h:m:s$ " è " $power$ " Watt"
```

E' possibile eseguire la formattazione personalizzata dei valori numerici utilizzando l'operatore FORMAT "<pattern>" operator.

Il pattern di formattazione è composto da un sotto-pattern positivo ed uno negativo, ad esempio "#,##0.00;(#,##0.00)". Ogni sotto-pattern ha un prefisso, una parte numerica ed un suffisso. Il sotto-pattern negativo è opzionale; se assente allora il sotto-pattern positivo viene utilizzato anche per rappresentare numeri negativi, che saranno preceduti dal simbolo '-' (nella maggior parte delle lingue). Quindi "0.00" è equivalente a "0.00;-0.00".

Il simbolo # rappresenta una cifra che non viene visualizzata se la cifra è zero. Il carattere 0 invece rappresenta una cifra sempre visualizzata. Per ulteriori informazioni sul formato dei pattern, consultare la documentazione della classe `java.text.DecimalFormat`¹⁸.

Il pattern può anche avere un prefisso di due caratteri seguito da ":", ad esempio "EN:#,##0.00", per specificare una lingua o nazione. In questo caso il pattern ed il formato risultante dipenderanno dalle regole della lingua o nazione specificata. Se il prefisso di localizzazione non è presente, verranno utilizzate le regole di localizzazione associate alla lingua specificata con il parametro Language in `hsyco.ini`.

Esempio:

```
$voltage : $rf = $voltage, $rf FORMAT "#.00"
```

¹⁸ <http://docs.oracle.com/javase/6/docs/api/java/text/DecimalFormat.html>

la variabile \$rf è impostata al valore di \$voltage, e formattata con due cifre decimali fisse.

Esempio:

```
$voltage : $rf = $voltage, $rf FORMAT "IT: #,00"
```

la variabile \$rf è impostata al valore di \$voltage, con due cifre decimali fisse, utilizzando le regole di localizzazione Italiane e la virgola come separatore decimale.

Variabili e Eventi

Le variabili possono essere utilizzate nelle espressioni di evento, confrontando il contenuto con valori costanti, con un'altra variabile, oppure confrontando lo stato di un dispositivo con il valore di una variabile.

Gli operatori di confronto sono:

= > < >= <= *=¹⁹

Il confronto viene effettuato tra numeri se entrambi gli operandi sono numerici, altrimenti tra testo, ignorando la differenza tra lettere minuscole e maiuscole. Se uno dei due operandi è un numero e l'altro è "off", il numero viene confrontato con 0.

Esempio:

```
HSYCOSTART : $contatore = 0
```

```
IO k.33 = 1 : $contatore + 1
```

```
$contatore : LOG = "Incremento il contatore al valore:"  
$contatore
```

```
$contatore = 10 : LOG= "Decima accensione della luce 33"
```

Un evento definito con il solo nome di variabile, senza operatore di confronto, costituisce un evento transitorio, che si verifica al cambiamento di stato della variabile, indipendentemente dal valore assunto.

¹⁹ L'operatore *= consente di verificare se la stringa a sinistra contiene la stringa di destra.

E' possibile verificare se una variabile è definita. La condizione:

```
$<varname> = ""
```

è vera se la variabile non è definita o è stata esplicitamente impostata a "".

Variabili Volatili e Persistenti

I valori impostati nelle variabili vengono normalmente persi nel momento in cui HSYCO viene riavviato. E' possibile rendere persistente una variabile semplicemente terminando il nome con il carattere !

```
$contatore!
```

è una variabile persistente, il cui valore viene mantenuto anche quando HSYCO viene riavviato. Le variabili persistenti sono molto utili, ma non è sempre necessario che una variabile sia persistente.

E' bene non utilizzare una variabile nella forma persistente se non necessario.

\$contatore e *\$contatore!* sono due variabili distinte.

Variabili Predefinite

EVENTS prevede alcune variabili predefinite:

```
$TIME$
$SUNSET$
$SUNRISE$
$TIME:HMS$
$DATE:YMD$
$DATE:DOW$
$POWER$
```

Le variabili predefinite si distinguono immediatamente da quelle normali perché il nome termina con il carattere \$.

\$TIME\$, \$SUNSET\$ e \$SUNRISE\$ sono utili in particolare per la gestione di eventi legati al tempo e all'orario di alba e tramonto.

Assumono il valore del momento attuale, del prossimo tramonto e della prossima alba, in un formato interno in secondi.

Inoltre, supportano un formato esteso:

`$TIME:HMS$`

`$DATE:YMD$`

consentendo così di ottenere l'ora attuale, i minuti e secondi, e il numero del giorno, mese e anno, individualmente o combinati in una stringa, con la possibilità di definire i caratteri di separazione.

La variabile:

`$DATE:DOW$`

restituisce il giorno della settimana in formato numerico, con 1 corrispondente a Lunedì e 7 a Domenica.

Esempio:

`$TIME:H$` restituisce solo l'ora, in formato a due cifre; `$TIME:M$` i minuti; `$TIME:S$` i secondi.

`$TIME:H:M:S$` restituisce un testo tipo *09:15:30*.

`$DATE:Y$` restituisce l'anno; `$DATE:M$` il mese; `$DATE:D$` il giorno.

`$DATE:D/M/Y$` restituisce un testo tipo *14/07/2011*.

Esempio:

`$POWER$`

restituisce il valore intero in Watt del consumo istantaneo, lo stesso dato visualizzato normalmente nell'interfaccia Web di HSYCO.

Esempio di alba:

`DAY OR HSYCOSTART: $sunset1h=$sunset$, $sunset1h - 3600, IO k.33 k.34 = 0, $lights = off`

La variabile `$tramonto-1h` conterrà un numero che rappresenta il tempo di un'ora prima del tramonto.

L'evento qui sotto farà accendere le luci dopo quest'orario:

`TIME > $sunset1h AND NOT $lights = on:
IO k.33 k.34 = 1, $lights = on`

Eventi

Il formato dell'evento deve essere obbligatoriamente composto dalla parola chiave del tipo di evento e dall'attributo associato, ad esempio l'indirizzo di un dispositivo, ed eventualmente dalla condizione.

I/O Server

IO

Evento richiamato al cambiamento di stato di qualunque data point di un I/O Server.

Event	Stato	Descrizione
IO name	transitorio	qualsiasi cambiamento di una variabile I/O
IO name = value IO name > value IO name < value IO name >= value IO name <= value	stabile	il valore è uguale, maggiore, maggiore uguale, minore, minore uguale del valore dato i nomi e i valori delle variabili di I/O sono legate al tipo di I/O Server. I valori possono essere numeri o stringhe

Parametri:

name - il nome del data point dell'ingresso o dell'uscita. A seconda del tipo di server, il formato cambia ma generalmente rappresenta il nome del server, come dichiarato nel parametro ioServers in hsyco.ini, seguito da un punto e dal nome della porta di ingresso/uscita. Consultare le Application Notes per ulteriori dettagli.

Esempi:

```
IO tempmeter.indoor > 25.5  
IO contacts.c1 = 1
```


IOSTART

Evento che si verifica quando la connessione con un server I/O è stata stabilita correttamente. E' possibile associare questo tipo di evento ad azioni che eseguano comandi sul server di I/O.

Evento	Stato	Descrizione
IOSTART id	transitorio	Evento che si verifica quando viene avviato il thread di monitor di ciascun server I/O - la prima volta per server all'inizio dell'esecuzione di HSYCO e successivamente dopo ogni riavvio del thread di monitor, ad esempio se si verificano errori di comunicazione.

Parametri:

id - numero progressivo dell'interfaccia, a partire da 0, in base alla lista di id definiti dal parametro *ioServers* in *hsyco.ini*; se è presente una sola interfaccia è possibile ometterlo.

Esempi:

```
IOSTART: IO contacts.c1 = 1  
IOSTART 2: IO relays.door = 1
```


DMX

DMXSTART

Evento che si verifica durante l'avvio dei thread di monitor di ciascun bus DMX, quindi una volta per bus all'inizio dell'esecuzione di HSYCO, ma anche in seguito ad ogni riavvio del thread.

Evento	Stato	Descrizione
DMXSTART busid	transitorio	evento transitorio che si verifica quando HSYCO effettua la connessione al gateway DMX. HSYCO stabilisce la connessione al gateway durante l'avviamento. Se il gateway viene spento e riacceso, o se si verificano errori di comunicazione, HSYCO ristabilisce automaticamente la connessione quando possibile, richiamando questo evento in caso di successo

Parametri:

busid -busid è utilizzato per distinguere più gateway DMX. Il primo gateway ha indirizzo 0. Il secondo ha indirizzo 1 ecc. Se presente un solo gateway, busid può essere omissso.

Esempi:

```
DMXSTART: DMX 100-200 = OFF  
DMXSTART 1: DMX 1100-1200 = OFF
```


DMX

Evento richiamato al cambiamento di stato sui bus DMX.

Evento	Stato	Descrizione
DMX address	transitorio	qualunque variazione del valore dell'indirizzo
DMX address = ON	stabile	indirizzo con valore > 0
DMX address = OFF	stabile	indirizzo con valore = 0
DMX address = value DMX address > value DMX address < value DMX address >= value DMX address <= value	stabile	il valore associato all'indirizzo è uguale, maggiore, maggiore o uguale, minore, minore o uguale al valore indicato. Sono validi valori compresi tra 0 e 255

Parametri:

address - se è presente un solo gateway DMX, address sarà un numero compreso tra 1 e 512, come previsto dallo standard DMX-512. Se sono presenti più gateway, al numero di base deve essere sommato 1000 per il secondo gateway, 2000 per il terzo ecc. Ad esempio, 2100 indica l'indirizzo 100 sul bus DMX gestito dal terzo gateway.

Esempi:

```
DMX 100 = ON
DMX 1200 = OFF
DMX 135 = OFF OR DMX 100 > 50
DMX 41 <= $LEVEL
```


Telecamere

CAMERA

Questo evento diventa vero nel momento in cui HSYCO sta registrando un video da una telecamera.

Viene richiamato in seguito ad un evento notificato da una telecamera inviando al Web server di HSYCO una richiesta tipo:

```
http://192.168.0.50/x/camerarec?camera=<name>
```

```
http://192.168.0.50/x/camerarec?camera=<name>&zone=<id>
```

La registrazione può essere attivata anche richiamando il metodo Java CameraRecTrigger() o tramite le azioni CAMERAREC o CAMERARECFULL in EVENTS.

Evento	Stato	Descrizione
CAMERA name	stabile	la telecamera inizia a registrare a seguito di un evento di motion rilevato e notificato dalla telecamera, oppure dopo un comando di registrazione
CAMERA name = zone	transitorio	evento transitorio che si verifica quando una arriva una richiesta di registrazione con l'informazione della zona

Parametri:

name - id della telecamera

zone - id della zona.

Esempi:

CAMERA entrance

CAMERACOMMAND


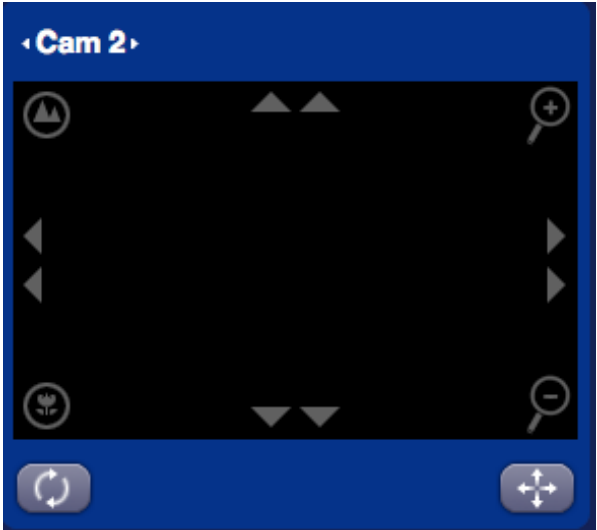







Evento richiamato in seguito ad un comando PTZ richiesto dall'interfaccia Web di HSYCO su una telecamera. Può essere utilizzato per eseguire qualunque tipo di azione nel caso l'utente clicchi sulle aree attive dell'interfaccia.

Evento	Stato	Descrizione
CAMERACOMMAND name = function	transitorio	Se l'evento corrispondente a funzione/azione PTZ è presente in EVENTS il comando PTZ standard associato al driver non viene eseguito

Parametri:

name - id della telecamera

function - può assumere uno dei valori elencati nella tabella seguente.

	function	
	focusfar	
	focusnear	
	zoomtele	
	zoomwide	
	moveleft	
	moveright	
	moveup	
	movedown	
Center	movestop	

Esempi:

```
CAMERACOMMAND entrance = focusfar
```


Antintrusione

SECURITY

Questo evento è deprecato e sarà eliminato nelle versioni successive.

Le centrali antintrusione *Paradox EVO*, *Bentel KYO320* e *Tecnoalarm*, direttamente supportate da HSYCO, generano molti eventi I/O proprietari. Per semplicità di programmazione vengono anche generati alcuni eventi di tipo SECURITY associati a inserimento, disinserimento e allarme.

Evento	Stato	Descrizione
SECURITY id = ALARM n	stabile	stato di allarme area/partizione/programma N
SECURITY id = ON n	stabile	area/partizione/programma N inserito
SECURITY id = OFF n	stabile	area/partizione/programma N disinserito

Parametri:

- id - identifica la centrale antintrusione; corrisponde agli id definiti nei rispettivi file di configurazione. id non può essere omesso
- n - numero intero positivo che identifica il numero dell'area (Paradox), della partizione (Bentel) o del programma (Tecnoalarm). n non può essere omesso.

Esempi:

```
SECURITY evo = ALARM 1  
SECURITY kyo = ON 2  
SECURITY kyo = OFF 2
```


IRTrans

IR

Evento che si verifica quando un IRTrans riceve o emette un comando IR presente nel database di comandi memorizzati.

Evento	Stato	Descrizione
IR name = command	transitorio	evento transitorio che si verifica quando un IRTrans riceve o emette un comando IR presente nel database di comandi memorizzati

Parametri:

name - identificativo dell'IRTrans e deve corrispondere a uno dei nomi presenti in hsyco.ini

command - compare nel formato <telecomando>.<comando>; ossia è costituito dal nome che identifica il telecomando seguito da punto e dal nome del comando.

Esempi:

```
IR theater = dvd.play
```


Squeezebox

MUSIC

Evento richiamato al cambiamento di stato di un player Squeezebox. E' un evento stabile che rappresenta lo stato corrente del player, che può essere ON/OFF/PLAY/PAUSE.

Evento	Stato	Descrizione
MUSIC address = ON	stabile	player acceso
MUSIC address = OFF	stabile	player spento
MUSIC address = PLAY	stabile	player acceso e in play
MUSIC address = PAUSE	stabile	player acceso e in pausa

Parametri:

address - numero che indica il player, a partire da 0 per il primo player, in base all'elenco di id definiti dal parametro *slimPlayers* in *hsyco.ini*.

Esempi:

MUSIC 0 = ON

MUSIC 1 = PLAY

PBX

PBX

Evento relativo alla notifica di chiamate VoIP gestite dalla centrale telefonica interfacciata a HSYCO.

Evento	Stato	Descrizione
PBX from = to	transitorio	si verifica nel momento in cui viene composto il numero, anche se alla chiamata non segue una risposta o se il numero chiamato non esiste.

Parametri:

from - indica il numero del chiamante

to - indica il numero chiamato.

Esempi:

Una chiamata dal numero 22 al numero 44 accende la luce k.33:

PBX 22 = 44 : IO k.33 = 1

Servizi di Rete

LOCATION

Se il servizio di localizzazione è attivo, l'evento è richiamato quando viene rilevata una variazione nell'associazione di un client rispetto agli Access Point Wi-Fi. E' un evento stabile che rappresenta lo stato corrente della connessione e la localizzazione dei clients Wi-Fi loggati all'interno dell'interfaccia Web.

Evento	Stato	Descrizione
LOCATION address = zone	stabile	quando un dispositivo si collega all'access point identificato con il nome della zona
LOCATION address = ON	stabile	quando un dispositivo si collega alla rete Wi-Fi
LOCATION address = OFF	stabile	quando un dispositivo si scollega dalla rete Wi-Fi

Parametri:

address - MAC address del dispositivo Wi-Fi. E' costituito da 6 coppie di byte, con rappresentazione esadecimale, separati dal carattere ":".

Esempi:

```
LOCATION aa:01:33:e4:41:00 = entrance
LOCATION aa:01:33:e4:41:00 = ON
LOCATION aa:01:33:e4:41:00 = OFF
```

PING

Questo evento viene richiamato in seguito ad un'azione di PING, che verifica se uno o più host IP, identificati dall'hostname o dall'indirizzo IP sono raggiungibili entro un periodo di tempo definito in millisecondi (o utilizzando un timeout di default impostato a 200 ms).

Evento	Stato	Descrizione
PING hostname = ON	transitorio	il dispositivo è stato raggiunto a seguito di un'azione di PING

Evento	Stato	Descrizione
PING hostname = OFF	transitorio	il dispositivo non è stato raggiunto a seguito di un'azione di PING

Parametri:

hostname - nome del dispositivo IP o indirizzo IP, come definito nella corrispondente azione PING

Esempi:

```
TIME: PING 192.168.1.1
PING 192.168.1.1 = OFF : LOG = "il router è off-line"
```

URL

Questo evento viene richiamato in seguito ad un'azione URL, che esegue una richiesta HTTP o HTTPS verso un url. E' possibile confrontare il valore numerico del codice di stato della risposta, oppure verificare la presenza di una stringa all'interno del corpo della risposta.

Evento	Stato	Descrizione
URL "url" = status_code	transitorio	il valore numerico del codice di stato della risposta è uguale a status_code
URL "url" *= "string"	transitorio	la stringa specificata è presente nel corpo della risposta

Parametri:

url - l'url completo, compreso il prefisso obbligatorio http:// o https:// e la query string opzionale

status_code - il valore numerico del codice di stato della risposta; una richiesta eseguita con successo genera normalmente una risposta con stato 200

string - l'evento è vero se la stringa specificata è presente nel corpo della risposta.

Esempi:

```
TIME 0800: URL GET "http://10.0.0.5/cgi-bin/cmd.sh?id=relay1"
URL "http://10.0.0.5/cgi-bin/cmd.sh?id=relay1" :
LOG = "relay opened"
```


Timer e Scheduler

TIMER

Evento stabile che viene richiamato nel momento in cui un timer o uno scheduler viene attivato o disattivato.

Evento	Stato	Descrizione
TIMER name = ON	stabile	si verifica quando il timer o lo scheduler è attivo (stato ON)
TIMER name = OFF	stabile	si verifica quando il timer o lo scheduler non è attivo (stato OFF)

Parametri:

name - l'identificativo del timer definito dal parametro *Timers* nel file hsyco.ini, o il nome della regola dello scheduler.

Esempi:

```
TIMER irrigation = ON : IO k.33 k.34 = 1  
TIMER irrigation = OFF : IO k.33 k.34 = 0
```


Leak Detector

LEAK

Evento che si verifica a seguito di un'azione LEAK quando l'Intelligent Leak Detector rileva una condizione di perdita, o quando la perdita viene cancellata.

Per maggiori dettagli vedere anche l'azione LEAK.

Evento	Stato	Descrizione
LEAK name	transitorio	evento che si verifica su qualunque cambiamento di stato del leak detector
LEAK name = ON	stabile	rilevazione perdita
LEAK name = OFF	stabile	condizione perdita off

Parametri:

name - nome del leak detector. Per implementare più rilevatori di perdita indipendenti utilizzare nomi differenti.

Esempi:

```
TIME: IO modbus.2.768 = readholdingregisters:uint,  
      LEAK water = IO modbus.2.768  
LEAK water = ON : MAIL john@example.com = hsyco@example.com  
                  "Messaggio urgente da HSYCO" "Perdita Rilevata"  
LEAK water = OFF : MAIL john@example.com = hsyco@example.com  
                   "Messaggio da HSYCO" "Reset Perdita"
```


Eventi Interni

HSYCOSTART

Evento relativo all'avvio di HSYCO che si verifica dopo l'inizializzazione del server HTTP, ma prima dell'esecuzione dei vari thread di monitor e del Webserver.

Evento	Stato	Descrizione
HSYCOSTART	transitorio	si verifica una volta sola alla partenza di HSYCO

Esempi:

```
HSYCOSTART : $count = 0
```

USER

Evento richiamato in seguito alla pressione dei tasti creati nell'interfaccia Web con gli oggetti (user), (usermini), (usermicro), (userimage) o (userrgb), o tramite l'azione USER o il metodo Java user().

Evento	Stato	Descrizione
USER name = param	transitorio	il formato richiede una corrispondenza tra il valore del nome e quello del parametro
USER name	transitorio	si verifica indipendentemente dal valore del parametro id

Parametri:

name - il campo <name> dell'oggetto (user)

param - il campo <id> dell'oggetto (user).

Esempi:

Accende la luce k.33, 10 secondi dopo la pressione del tasto:

```
USER timer : PROGRAMTIMER test = SET 10
PROGRAMTIMER test : IO k.33 = 1
```


PROGRAMTIMER

Evento richiamato in seguito all'attivazione di un program timer.

Evento	Stato	Descrizione
PROGRAMTIMER name	transitorio	si verifica allo scadere del program timer

Parametri:

name - l'identificativo del program timer.

Esempi:

Accende la luce k.33, 10 secondi dopo la pressione del tasto:

```
USER timer : PROGRAMTIMER test = SET 10
PROGRAMTIMER test : IO k.33 = 1
```

TIME

Evento transitorio richiamato ogni 60 secondi. HSYCO sincronizza l'esecuzione all'inizio del minuto, cioè l'evento si verifica al secondo 00 di ogni minuto. Può essere utilizzato anche come evento stabile, confrontando l'ora corrente con quella specificata nel formato HHMM.

Evento	Stato	Descrizione
TIME	transitorio	viene attivato all'inizio di ogni minuto
TIME = hhmm TIME > hhmm TIME < hhmm TIME >= hhmm TIME <= hhmm	stabile	confronta l'ora corrente con quella specificata nel formato HHMM

Esempi:

```
TIME : CAMERAREC entrance = 60
TIME = 0930 : IO k.33 = 0
```


DAY

Evento stabile che risulta vero al momento dell'alba, calcolato in base ai valori di latitudine e longitudine impostati in hsyco.ini e ai parametri SunriseOffsetMinutes e SunsetOffsetMinutes.

Evento	Stato	Descrizione
DAY	stabile	si verifica all'alba; la condizione rimane vera fino al tramonto

Esempi:

DAY : IO k.33 k.34 k.35 = 0

NIGHT

Evento stabile che risulta vero al momento del tramonto, calcolato in base ai valori di latitudine e longitudine impostati in hsyco.ini e ai parametri SunriseOffsetMinutes e SunsetOffsetMinutes.

Evento	Stato	Descrizione
NIGHT	stabile	si verifica al tramonto; la condizione rimane vera fino all'alba

Esempi:

NIGHT: IO k.33 k.34 k.35 = 1

SUNAZIMUTH

Evento richiamato ad ogni variazione della parte intera dell'angolo di azimuth del sole (l'angolo compreso tra la direzione del Nord e quella Sud all'orizzonte), calcolata in base ai valori di latitudine e longitudine impostati in hsyco.ini e ai parametri SunriseOffsetMinutes e SunsetOffsetMinutes.

Evento	Stato	Descrizione
SUNAZIMUTH = value SUNAZIMUTH > value SUNAZIMUTH < value SUNAZIMUTH >= value SUNAZIMUTH <= value	stabile	risulta vero se la condizione di confronto è soddisfatta. Per il confronto tra gli angoli i decimali vengono ignorati

Parametri:

value - numero corrispondente ai gradi dell'angolo, compreso tra 0 e 359.

Esempi:

`SUNAZIMUTH > 180 AND SUNAZIMUTH < 220 : IO k.54 = 0`

SUNELEVATION

Evento richiamato ad ogni variazione della parte intera dell'angolo di elevazione del sole (l'angolo compreso tra l'orizzonte e la posizione del Sole), calcolata in base ai valori di latitudine e longitudine impostati in hsyco.ini e ai parametri SunriseOffsetMinutes e SunsetOffsetMinutes.

Evento	Stato	Descrizione
SUNELEVATION = value SUNELEVATION > value SUNELEVATION < value SUNELEVATION >= value SUNELEVATION <= value	stabile	risulta vero se la condizione di confronto è soddisfatta. Per il confronto tra gli angoli i decimali vengono ignorati

Parametri:

value - numero corrispondente ai gradi dell'angolo, compreso tra -90 e 90.

Esempi:

```
SUNELEVATION > 45 : IO k.55 = 0
```

POWER

Evento richiamato a seguito di un cambiamento del livello di carico di potenza, impostato utilizzando il metodo Java powerSet() o l'azione POWER.

Evento	Stato	Descrizione
POWER	transitorio	si verifica quando viene rilevato qualunque cambiamento del livello di potenza
POWER > power POWER < power POWER >= power POWER <= power	stabile	si verifica quando viene rilevato un cambiamento del livello di potenza e la condizione è soddisfatta. La potenza è definita in Watt

Esempi:

```
POWER > 2000 : IO k.70 = 0
POWER : LOG "NEW POWER:" $POWER$
```


Azioni

Il formato dell'azione deve essere composto dalla parola chiave del tipo di evento e dall'attributo associato, ossia l'oggetto o lo stato relativo.

I/O Server

IO

Azione che imposta variabili a I/O Server o data point.

Azione	Descrizione
IO name = value	imposta la variabile ad un valore. I valori dipendono dal tipo di server I/O
IO name = FLIP	inverte l'uscita di un data point di I/O. Se lo stato dell'uscita è 0, il nuovo stato sarà 1; se diverso da 0 il nuovo stato sarà 0
IO name1 = IO name2	imposta l'uscita dell'interfaccia I/O con nome1 al valore attuale dell'ingresso o uscita dell'interfaccia I/O con nome2

Parametri:

name - identifica l'interfaccia e lo specifico ingresso o uscita. Il formato dipende dal tipo di interfaccia gestita, ma generalmente è composto dal nome dell'interfaccia, come dichiarato nel parametro *ioServers* in *hsyco.ini*, seguito da un punto e dal nome assegnato all'ingresso/uscita

value - è possibile utilizzare più parole non racchiuse tra virgolette, stringhe racchiuse tra virgolette e variabili, che verranno aggiunti al valore della stringa

Esempi:

```
IO contacts.c1 contacts.c3= 1
IO sensors.c1 = FLIP
IO contacts.o1 contacts.o2 = IO sensors.flood
```


DMX

Azione che imposta il valore dei canali sul bus DMX-512.

Azione	Descrizione
DMX address = ON	imposta l'indirizzo DMX al valore precedente l'ultimo comando di OFF
DMX address = OFF	imposta l'indirizzo DMX al valore = 0
DMX address = FLIP	provoca un'inversione di stato dell'indirizzo DMX (se lo stato è ON esegue il comando OFF e viceversa). Quando sono indicati più indirizzi, se lo stato precedente di almeno uno dei canali è diverso da OFF, il comando OFF viene inviato a tutti i canali, altrimenti viene inviato il comando ON
DMX address = value	assegna un valore all'indirizzo DMX Sono validi valori compresi tra 0 e 255
DMX addressX = DMX addressY	il dispositivo DMX con indirizzo X assume il valore attuale del dispositivo DMX con indirizzo Y
DMX address = MERGE	l'indirizzo sul bus DMX OUT del gateway segue i valori dell'indirizzo del bus DMX IN
DMX address = UNMERGE	ritorna al funzionamento normale

Parametri:

address - numero compreso tra 1 e 512, come previsto dallo standard DMX-512. Se sono presenti più gateway, al numero di base deve essere sommato 1000 per il secondo gateway, 2000 per il terzo ecc. Ad esempio, 2100 indica l'indirizzo 100 sul bus DMX gestito dal terzo gateway. E' possibile specificare blocchi di canali contigui, indicandoli nel formato da-a, ad esempio 100-150.

Esempi:

```
DMX 100 101 102 = ON
DMX 100-120 200-220 = DMX 40
DMX 33 = MERGE
```


Telecamere

CAMERA

Azione che abilita o disabilita temporaneamente una telecamera. Quando la telecamera è disabilitata vengono disattivate anche le funzioni di registrazione.

Azione	Descrizione
CAMERA name = ON	attiva la visualizzazione della telecamera indicata
CAMERA name = OFF	disattiva la visualizzazione della telecamera indicata
CAMERA name = RECON	abilita la registrazione di immagini della telecamera indicata
CAMERA name = RECOFF	disabilita la registrazione di immagini della telecamera indicata

Parametri:

name - id della telecamera. E' possibile specificare più nomi di telecamere, separati da spazio.

Esempi:

```
CAMERA entrance = ON
```

CAMERAREC

Azione che attiva la registrazione dei frame acquisiti da una telecamera

Azione	Descrizione
CAMERAREC name = sec	attiva la registrazione di immagini della telecamera indicata per il numero di secondi impostato

Parametri:

name - id della telecamera. E' possibile specificare più nomi di telecamere, separati da spazio.

Esempi:

```
CAMERAREC openspace = 30
```


CAMERARECFULL

Azione che attiva la registrazione dei frame acquisiti da una telecamera ignorando l'opzione *DroppedFrames* specificata nel file *hsyco.ini*, in modo tale che tutte le frame acquisite dalla telecamere durante la registrazione vengano registrate in modo continuativo e senza salti.

Azione	Descrizione
CAMERARECFULL name = sec	registra per il numero di secondi impostato

Parametri:

name - id della telecamera. E' possibile specificare più nomi di telecamere, separati da spazio.

Esempi:

```
CAMERARECFULL openspace = 30
```


Porte di Comunicazione Seriale

COMM

Invia una sequenza di byte alla porta seriale specificata.

Azione	Descrizione
COMM name = hexbytes	invia una sequenza di byte alla porta seriale

Se entrambi i parametri verboseLog = true e userLog = true, viene scritta l'intera sequenza di byte inviati sul file di log

Parametri:

name - nome della porta di comunicazione e deve corrispondere a uno dei nomi presenti in hsyco.ini

hexbytes - stringa con la rappresentazione esadecimale della sequenza di byte da inviare. Sono ammesse stringhe di testo delimitate con “” e variabili. E' possibile concatenare più stringhe e variabili.

Esempi:

```
COMM serialport = FE03C9104B27  
COMM serialport = FE $body 27
```


IRTrans

IR

Azione che consente di inviare comandi infrarossi (IR) attraverso l'IRTrans gestito da HSYCO.

Azione	Descrizione
IR name = command	Il codice IR deve essere memorizzato nel database locale dell'IRTrans, oppure se la directory hsyco/ir contiene il file .ccfhex corrispondente al nome del database, viene utilizzata la stringa CCF invece di inviare il comando contenuto nel database interno dall'IRTrans

Parametri:

name - identificativo dell'IRTrans come definito nel parametro IRTrans presente in hsyco.ini

command - compare nel formato <telecomando>.<comando>; ossia è costituito dal nome che identifica il telecomando seguito da punto e dal nome del comando.

Esempi:

```
IR theater = dvd.play
```


Squeezebox

MUSIC

Azione che consente di inviare qualunque comando ad un player Squeezebox.

Azione	Descrizione
MUSIC address = ON	accende il player indicato
MUSIC address = OFF	spegne il player indicato
MUSIC address = PLAY	avvia l'esecuzione del player indicato

Parametri:

address - numero che indica il player, a partire da 0 per il primo player, basato sull'elenco ordinato del parametro slimPlayers in hsyco.ini.

Esempi:

```
MUSIC 0 = ON  
MUSIC 1 = PLAY
```


Public Announcement

AUDIO

L'azione AUDIO è utilizzata per riprodurre annunci vocali text-to-speech o file audio registrati.

Gli annunci vocali possono essere inviati al browser Web, oppure tramite l'uscita audio line-out del server, attraverso l'altoparlante interno o l'uscita audio delle telecamere Axis e l'altoparlante dei telefoni SNOM o dei dispositivi PA. Per maggiori dettagli consultare l'Appendice Public Announcement.

Azione	Descrizione
AUDIO to = FILE filename	riproduce un file audio pre-registrato
AUDIO to = VOICE:voicename message	traduce un messaggio di testo in un messaggio vocale e lo riproduce

Parametri:

to - destinazione messaggio vocale (vedi tabella sotto)

filename - il percorso del file audio, in relazione alla directory principale di HSYCO

voicename - il nome della voce di text-to-speech; per maggiori dettagli consultare l'Appendice Public Announcement

message - il messaggio di testo da convertire; è possibile utilizzare stringhe multiple che verranno automaticamente aggiunte alle precedenti

Destinazione audio	Descrizione
speaker	uscita audio line-out del server
web	uscita audio del browser Web (funzione supportata solo su Firefox e Chrome)

Destinazione audio	Descrizione
axis@<camera id>	messaggio vocale inviato a una telecamera Axis. Impostare camera id in base al nome definito per il parametro Cameras in hsyco.ini
snom@<ip:port>	messaggio vocale inviato ai telefoni SNOM e ai dispositivi di public announcement, all'indirizzo IP multicast e alla porta specificata (l'indirizzo IP e la porta devono essere configurati come indirizzi multicast su ogni telefono)

Esempi:

TIME : audio speaker = file "audio/gong.mp3"

Il file audio gong.mp3 viene riprodotto ogni minuto attraverso l'uscita audio line-out del server o tramite un altoparlante interno.

TIME : audio snom@239.255.255.245:5555 = file "audio/gong.mp3"

Come sopra, ma il messaggio vocale viene riprodotto ogni minuto su tutti i telefoni SNOM registrati all'indirizzo multicast 239.255.255.245 e alla porta 5555.

TIME : audio axis@cam1 = file "audio/gong.mp3"

Come sopra, ma il messaggio vocale viene riprodotto attraverso la telecamera Axis con id "cam1".

TIME : audio axis@cam1 = "voice:en" "the time is " \$time:h:m\$

Converte un messaggio di testo in messaggio vocale utilizzando la voce "en" del text-to-speech, e lo riproduce attraverso una telecamera Axis.

Mail

MAIL

Invia una email. HSYCO SERVER invia una email o direttamente al dominio di mail server del destinatario se il parametro SmtplibName non è stato dichiarato in hsyco.ini, oppure utilizzando un account email specifico con autenticazione dell'utente e traffico criptato se il server STMP e i parametri dell'account sono stati definiti. E' necessario verificare che il mail server destinatario del messaggio accetti e consideri valida la mail proveniente dall'indirizzo specificato come mittente, inviata dall'indirizzo IP pubblico di HSYCO SERVER.

Questo metodo non ritenta di inviare il messaggio se il mail server di destinazione non è disponibile quando il metodo viene chiamato.

Il corpo della mail può essere un semplice testo o un messaggio multi-part, composto da immagini live o registrate delle telecamere e da file allegati.

Azione	Descrizione
MAIL to = from subject body	invia email con immagini delle telecamere. Il corpo della mail è l'ultimo attributo definito. E' possibile inviare lo stesso messaggio e più destinatari

Parametri:

to - indirizzo email del destinatario. E' possibile scrivere una lista di indirizzi separati da spazi ed usare variabili. E' anche possibile specificare il nome del server SMTP o l'indirizzo IP aggiungendo alla fine dell'indirizzo email del destinatario :<nome server name o indirizzo>.

Ad esempio: *john@example.com:192.168.1.1*

from - un solo indirizzo email valido del mittente. E' possibile usare una variabile

subject - una singola stringa o variabile contenente l'oggetto del messaggio

body - il corpo del messaggio. Per inviare una mail di testo, digitare la stringa di testo tra virgolette. Per inviare un'immagine, aggiungere la stringa nel seguente formato:

"cam:<cameraname>[:<seconds_back>]". Ad esempio, "cam:door" invia un frame live dalla telecamera chiamata "door"; "cam:door:2" invia un frame che era stata registrato due secondi prima dell'ultimo frame; "cam:door:0" invia l'ultimo frame registrato. Per inviare file come allegati, utilizzare il seguente formato: "file:<file name>". Il percorso del file è relativo alla directory root di HSYCO.

Esempi:

```
TIME : $BODY = "Energy consumption for " $DATE:y/m/d$ " at "
$TIME:h:m:s$ " is " $power$ " Watt"
$BODY : MAIL email1@hsyco.com email2@hsyco.com =
hsyco@hsyco.net $BODY
```

In questo esempio, viene inviato una mail ogni minuto, con il carico di potenza misurato, a due destinatari.

```
SECURITY = ALARM : CAMERAREC ingr = 30, PROGRAMTIMER rec = 40
PROGRAMTIMER rec : MAIL email1@hsyco.com = hsyco@hsyco.net
"Camera ingr" "cam:ingr" "cam:ingr:0" "cam:ingr:15"
```

Dopo 40 secondi dalla segnalazione di allarme viene inviata una email all'indirizzo email1@hsyco.com con indirizzo mittente hsyco@hsyco.net, con tre immagini: la prima sarà una immagine in diretta ripresa al momento dell'invio della mail, la seconda sarà l'ultima immagine registrata, la terza l'immagine registrata circa 15 secondi prima dell'immagine precedente.

```
TIME = 0000: MAIL email1@hsyco.com = hsyco@hsyco.net "Sent
from HSYCO" "Energy daily report." file:data/energy.csv
```

A mezzanotte viene inviata una email con il file energy.csv in allegato, contenuto nella sotto-directory data di HSYCO

Log

LOG

Azione che consente di generare un messaggio personalizzabile nel file di log.

Azione	Descrizione
LOG = text	Sono ammesse stringhe di testo delimitate con "", stringhe senza doppi apici se non contengono caratteri speciali, variabili. E' possibile concatenare più stringhe e variabili

Esempi:

```
LOG = log message example
```

```
LOG = "Time: " $TIME:H-M-S$ " DATE: " $DATE:D/M/Y$
```


Servizi di Rete

PING

Verifica che uno o più host, identificati dall'hostname o dall'indirizzo IP, siano raggiungibili entro un timeout definito in millisecondi (o utilizzando un timeout di default di 200 ms). Questo test genera eventi di PING come PING <hostname> = ON se l'host è raggiungibile, o PING <hostname> = OFF se l'host non è raggiungibile.

Azione	Descrizione
PING hostname	verifica che l'hostname sia raggiungibile entro un timeout di 200 ms
PING hostname = time	verifica che l'hostname sia raggiungibile entro il timeout definito

Parametri:

hostname - nome del dispositivo IP o indirizzo IP

time - timeout di risposta, in millisecondi

Esempi:

```
TIME : PING 192.168.1.1  
TIME : PING 192.168.1.200 192.168.1.201 = 1000
```

URL

Questa azione esegue una richiesta HTTP o HTTPS di tipo GET o POST verso l'url specificato.

E' supportata l'autenticazione standard del protocollo HTTP, per cui è possibile eseguire le richieste specificando user e password.

Le risposte vengono verificate utilizzando il corrispondente evento URL.

Evento	Descrizione
URL GET "url"	esegue una richiesta di tipo GET senza autenticazione
URL GET "user:password" "url"	esegue una richiesta di tipo GET con autenticazione
URL POST "url" "content_type" "data"	esegue una richiesta di tipo POST senza autenticazione
URL POST "user:password" "url" "content_type" "data"	esegue una richiesta di tipo POST con autenticazione

Parametri:

url - l'url completo, compreso il prefisso obbligatorio http:// o https:// e la query string opzionale

user:password - user e password per l'autenticazione della richiesta

content_type - il descrittore MIME del tipo di contenuto inviato

data - solo per richieste di tipo POST, il corpo della richiesta preventivamente URL-encodato.

Esempi:

```
HSYCOSTART: URL POST "http://10.0.0.5/cgi-bin/cmd.sh" "text/xml" "%3C%3Fxml%20version%3D%22..."
```

```
TIME 0800: URL GET "usr1:qi3qw" ""http://10.0.0.5/cgi-bin/cmd.sh?id=relay1"
```

```
URL "http://10.0.0.5/cgi-bin/cmd.sh?id=relay1" :  
LOG = "relay opened"
```


Data Logger

Azione che consente di generare statistiche e file CSV a partire dalla variazione di un valore. Un data logger può essere utilizzato, ad esempio, per generare grafici e/o per raccogliere dati in un file CSV.

Per ulteriori dettagli consultare il capitolo Data Logger nelle Appendici di questo manuale.

DATALOGGER

Aggiorna ed esegue operazioni su un data logger.

Azione	Descrizione
DATALOGGER name = value	Fornisce un nuovo valore da processare al data logger. Attiva anche la procedura di refresh.
DATALOGGER name = CLEAR	Cancella i dati raccolti dal data logger.
DATALOGGER name = FILE LOG filename [TIMESTAMP]	Aggiunge una riga contenente l'ultimo valore acquisito al file specificato (filename) utilizzando il formato CSV. Se l'opzione TIMESTAMP è specificata, anche il timestamp corrispondente al momento di acquisizione del valore verrà aggiunto.
DATALOGGER name = FILE STAT filename	Aggiorna il file specificato (filename) con i valori attualmente raccolti dal data logger usando il formato CSV.

Parametri:

name - identificativo del data logger. E' possibile utilizzare una lista di nomi separati da spazi per indirizzare diversi data logger. Possono essere utilizzate variabili

filename - il percorso del file. E' possibile utilizzare stringhe multiple differenti, incluse variabili, che saranno concatenate per comporre il nome del file.

Esempio 1:

```
$value : DATALOGGER energy = $value,  
        DATALOGGER energy = FILE LOG hsyco/energylog.csv TIMESTAMP,  
        DATALOGGER energy = FILE STAT hsyco/energystat.csv
```

Esempio 2:

```
TIME : DATALOGGER temperature = IO ste.1  
TIME : DATALOGGER humidity = IO ste.2
```


Leak Detector

LEAK

L'Intelligent Leak Detector viene utilizzato per segnalare potenziali perdite di acqua o altre sostanze analizzando un generico contatore di flusso.

Il leak detector viene richiamato ogni volta che il contatore di flusso viene incrementato. Segnerà una perdita se il flusso misurato risulta relativamente costante oltre un determinato periodo di tempo.

Utilizza questo metodo a tuo rischio!

L'Intelligent Leak Detector utilizza un algoritmo di correlazione che distingue tra un flusso costante e un flusso irregolare. Un flusso relativamente costante oltre un certo periodo di tempo è ritenuto una potenziale perdita e genera un evento leak.

In alcune condizioni, anche in relazione al tipo di contatore di flusso utilizzato, questo algoritmo potrebbe fallire nel riconoscere una reale perdita, o potrebbe generare delle false segnalazioni di perdita.

La logica del leak detector sfrutta due parametri per impostare il livello di sensibilità e l'arco temporale.

Genera un messaggio di avviso se la deviazione del flusso rimane costante al di sotto di una soglia di deviazione per un determinato periodo di tempo.

E' possibile modificare i valori di default del leak detector per adattarsi a condizioni specifiche.

Azione	Descrizione
LEAK name = value	richiama il leak detector passando il valore di flusso aggiornato. E' supportato l'utilizzo di variabili o riferimenti ai data point IO

Azione	Descrizione
LEAK name = PERIOD t	<p>cambia il periodo di tempo base. Il valore di default è di 1200 secondi.</p> <p>Impostare il periodo 0 per disabilitare temporaneamente il leak detector.</p> <p>Il periodo di tempo può essere modificato in qualunque momento, anche quando il leak detector sta processando dati</p>
LEAK name = DEVIATION d	<p>cambia la percentuale di deviazione. d deve essere un numero tra 1 e 100. Il valore di default è 50.</p> <p>Valori più bassi possono provocare situazioni che attivano segnalazione di perdita solo se il flusso è molto costante nel tempo, diventando così meno suscettibile ai falsi allarmi, ma un valore troppo basso potrebbe anche perdere perdite effettive. Valori più alti possono invece generare falsi allarmi.</p> <p>Il valore di deviazione può essere modificato in qualunque momento</p>
LEAK name = CLEAR	<p>cancella lo stato interno del leak detector. In normali applicazioni, non c'è bisogno di richiamare questa funzione, poichè il leak detector cancella automaticamente il suo stato in relazione ai dati passati</p>

Parametri:

name - nome del leak detector. Utilizzare nomi differenti per implementare più rilevatori indipendenti.

Esempi:

```
TIME: IO modbus.2.768 = readholdingregisters:uint,
      LEAK water = IO modbus.2.768
LEAK water = ON : MAIL john@example.com = hsyco@example.com
                  "Messaggio urgente da HSYCO" "Perdita rilevata"
LEAK water = OFF : MAIL john@example.com = hsyco@example.com
                  "Messaggio urgente da HSYCO" "Reset Perdita"
```


Azioni Interne

WAIT

Azione che causa una pausa nell'esecuzione delle azioni per il numero di secondi impostato.

Azione	Descrizione
WAIT = s	causa una pausa nell'esecuzione delle azioni per il numero di secondi impostato. E' supportato anche il formato decimale

Esempi:

WAIT = 5

WAIT = 0.3

USER

Azione che richiama il metodo Java:

userCommand (String name, String param)

definito in user.java. Similmente può essere utilizzato come una chiamata del metodo in EVENTS, o come meccanismo di chiamata tra EVENTS e Java.

Azione	Descrizione
USER name = param	richiama il metodo Java definito in user.java: <i>userCommand (String name, String param)</i> e l'evento USER name = param. E' possibile omettere uno dei due parametri, ma non entrambi. Il parametro omissso sarà passato come stringa vuota

Parametri:

name - corrisponde al campo <name> dell'oggetto (user). E' possibile concatenare più stringhe e variabili per impostare il parametro name

param - corrisponde al campo <param> dell'oggetto (user). E' possibile concatenare più stringhe e variabili per impostare il parametro param.

Esempi:

```
USER hometheater = "on"
USER hometheater = "changed" $newvalue
```

POWER

Imposta la variabile di stato relativa alla potenza elettrica. Il valore di potenza viene mostrato nella barra di stato o di navigazione dell'interfaccia Web. Oltre a impostare il valore di potenza, questo metodo richiama anche l'esecuzione del metodo di callback powerEvent() e l'evento POWER in EVENTS.

Azione	Descrizione
POWER = power	imposta la variabile di stato della potenza elettrica a uno specifico valore, in Watt

Parametri:

power - la potenza in Watt.

Esempi:

```
POWER = IO meter.power
```

PROGRAMTIMER

Azione crea o cancella un program timer.

Azione	Descrizione
PROGRAMTIMER name = SET s	imposta un program timer con esecuzione dopo il numero di secondi indicato. Se è già impostato un programtimer con lo stesso nome, l'azione viene ignorata
PROGRAMTIMER name = CLEAR	cancella un program timer impostato

Azione	Descrizione
PROGRAMTIMER name = RESET s	è analogo a SET, ma se è già impostato un program timer con lo stesso nome il timer viene reimpostato per l'esecuzione dopo il numero di secondi indicato
PROGRAMTIMER name = REPEAT s	imposta un program timer con esecuzione ripetuta con un intervallo pari al numero di secondi indicato

Parametri:

name - nome del program timer.

Esempi:

```
SECURITY evo = ON 1 : PROGRAMTIMER presence = REPEAT 3600
SECURITY evo = OFF 1 : PROGRAMTIMER presence = CLEAR
```


UISET

Cambia gli attributi dinamici degli oggetti (text), (user), (link), e di tutti gli oggetti con la versione identificata.

Azione	Descrizione
UISET id.attr = value	id è il nome univoco che identifica l'oggetto specificato nel file index.hsm con l'estensione lid dopo il tipo dell'oggetto

Parametri:

id - id dell'oggetto

attr - nome dell'attributo; vedi tabella seguente

value - valore dell'attributo; vedi tabella seguente. E' possibile concatenare più stringhe e variabili per impostare il parametro value.

tipo oggetto	attributo	valore
project	page	imposta l'id al nome della directory del file index.hsm che si vuole controllare, o a "*" per controllare tutti i client Web. Impostando value all'id della pagina o a "menu" viene forzata la visualizzazione di quella pagina su tutti i browser che hanno il menu visibile nel momento in cui il comando viene eseguito. Se value è una stringa vuota il comando eseguito riporta alla pagina precedente.
	lock	imposta l'id al nome della directory del file index.hsm che si vuole controllare, o a "*" per controllare tutti i client Web. Impostando value all'id della pagina o a "menu" viene forzata la visualizzazione di quella pagina su tutti i browser che hanno il menù visibile nel momento in cui il comando viene eseguito, ma anche per i menù visualizzati successivamente all'esecuzione del comando. Inoltre viene bloccata la navigazione dell'utente. Se invece value è una stringa vuota la navigazione viene sbloccata.

tipo oggetto	attributo	valore
	pageback	imposta l'id al nome della directory del file index.hsm che si vuole controllare, o a "*" per controllare tutti i client connessi. Impostare value all'id della pagina per forzare i client che stanno visualizzando la pagina corrente a quella precedente. Impostare value all'id del pop-up per forzare i client che hanno quel pop-up aperto a chiuderlo
page	blink	<i>false</i> : tutti gli oggetti della pagina smettono di lampeggiare
background	img	nome del file personalizzato con l'immagine dell'oggetto salvato nella directory www/img, o l'URL completo (compreso di https:// iniziale) per mostrare un'immagine da un sito remoto
camera	camera	id della telecamera da visualizzare
camerapanel	visible	<i>true</i> : visibile; <i>false</i> : non visibile
	blink	<i>true</i> : lampeggio lento; <i>slow</i> : lampeggio lento; <i>fast</i> : lampeggio veloce; <i>false</i> : stop lampeggio
	pos	la posizione dell'oggetto
chart	visible	<i>true</i> : visibile; <i>false</i> : non visibile
	blink	<i>true</i> : lampeggio lento; <i>slow</i> : lampeggio lento; <i>fast</i> : lampeggio veloce; <i>false</i> : stop lampeggio
	pos	la posizione dell'oggetto
	type	tipo di grafico; valori ammessi: <i>bars</i> , <i>points</i> , <i>gauge</i> . Il chart di tipo gauge supporta solo i seguenti attributi: value, orientation, valuerange, bgcolor, barcolor, barborder; l'attributo valuerange è necessario per una corretta rappresentazione grafica del valore
	orientation	orientamento verticale o orizzontale dell'asse dei valori; valori ammessi: <i>vertical</i> , <i>horizontal</i>
	value	il valore, solo per chart di tipo gauge
	values	la serie di valori separati da virgola, per chart di tipo bars e points. E' possibile omettere alcuni valori della serie (es. values="1, 3, 2, ,4, , , 5")
	valuerange	valore minimo e massimo separati da virgola. Indicano il range del grafico. Es. "-100,100" (con orientation <i>vertical</i>) visualizzerà un grafico con valori da -100 a 100 sull'asse y

tipo oggetto	attributo	valore
	axislabels	etichette sull'asse x (se orientation è <i>vertical</i>) o sull'asse y (se orientation è <i>horizontal</i>). Non deve necessariamente corrispondere al numero di valori in <i>values</i> e può contenere label vuote (in tal caso vengono visualizzate solo le tacche)
	drawaxis	specifica se gli assi (comprese le tacche e le label) vengono disegnati o meno; valori possibili: <i>true</i> o <i>false</i>
	origin	sposta l'origine ad un valore diverso da 0 (ad esempio origin=3.2)
	axisoffset	l'offset dell'asse (y se vertical, x se horizontal) in percentuale o come indice della serie (es. "50%" o "3")
	baroffset	sposta le barre o i punti del numero di pixel specificato, rendendo più semplice la sovrapposizione di più chart
	notches	numero di tacche sull'asse x (se orientation è <i>horizontal</i>) o sull'asse y (se orientation è <i>vertical</i>)
	spacing	spaziatura in percentuale fra le barre (valido solo per la modalità <i>bars</i>)
	pointsize	dimensione dei "punti" (valido solo per la modalità <i>points</i>)
	vlabelsstyle	stile delle value label; valori possibili: <i>inside</i> , <i>outside</i> , <i>none</i>
	bgcolor	colore dello sfondo in codice HTML
	barcolor	serie di colori HTML, uno per ogni valore in <i>values</i>
	barborder	visibilità del bordo delle barre/punti; valori possibili: <i>true</i> o <i>false</i>
	axiscolor	colore HTML degli assi
	notchcolor	colore HTML delle tacche sugli assi
	labelcolor	colore HTML delle etichette sugli assi e sulle barre
container	visible	<i>true</i> : visibile; <i>false</i> : non visibile
	blink	<i>true</i> : lampeggio lento; <i>slow</i> : lampeggio lento; <i>fast</i> : lampeggio veloce; <i>false</i> : stop lampeggio
	pos	la posizione dell'oggetto
dlink	visible	<i>true</i> : visibile; <i>false</i> : non visibile
	blink	<i>true</i> : lampeggio lento; <i>slow</i> : lampeggio lento; <i>fast</i> : lampeggio veloce; <i>false</i> : stop lampeggio
	pos	la posizione dell'oggetto
	text	il testo visualizzato
	color	<i>r</i> : rosso, <i>g</i> : verde, <i>b</i> : blu, <i>gr</i> : grigio, <i>y</i> : giallo
image	visible	<i>true</i> : visibile; <i>false</i> : non visibile

tipo oggetto	attributo	valore
imagelink	blink	<i>true</i> : lampeggio lento; <i>slow</i> : lampeggio lento; <i>fast</i> : lampeggio veloce; <i>false</i> : stop lampeggio
	pos	la posizione dell'oggetto
	img	nome del file contenuto nella directory <code>www/img</code> , o l'URL completo (compreso di <code>https://</code> iniziale) per mostrare un'immagine da un sito remoto
	text	il testo visualizzato
input	visible	<i>true</i> : visibile; <i>false</i> : non visibile
keypad	blink	<i>true</i> : lampeggio lento; <i>slow</i> : lampeggio lento; <i>fast</i> : lampeggio veloce; <i>false</i> : stop lampeggio
	pos	la posizione dell'oggetto
	text	il testo visualizzato
	color	il colore del testo; utilizzare i codici di colore standard HTML
	size	la grandezza dei caratteri in pixel
	width	larghezza, in pixel
	height	altezza, in pixel
	weight	lo spessore del testo; valori possibili: <i>lighter</i> , <i>normal</i> , <i>bold</i> , <i>bolder</i>
	style	lo stile del testo; valori possibili: <i>normal</i> , <i>italic</i> , <i>oblique</i>
link	visible	<i>true</i> : visibile; <i>false</i> : non visibile
linkmini	blink	<i>true</i> : lampeggio lento; <i>slow</i> : lampeggio lento; <i>fast</i> : lampeggio veloce; <i>false</i> : stop lampeggio
linkmicro	pos	la posizione dell'oggetto
	text	il testo visualizzato
	color	<i>r</i> : rosso, <i>g</i> : verde, <i>b</i> : blu, <i>gr</i> : grigio, <i>y</i> : giallo
marquee	visible	<i>true</i> : visibile; <i>false</i> : non visibile
	blink	<i>true</i> : lampeggio lento; <i>slow</i> : lampeggio lento; <i>fast</i> : lampeggio veloce; <i>false</i> : stop lampeggio
	pos	la posizione dell'oggetto
	text	il testo visualizzato
	color	il colore del testo; utilizzare i codici di colore standard HTML
	size	la grandezza dei caratteri, in pixel
	width	la larghezza del campo di testo, in pixel
	align	l'allineamento del testo; valori possibili: <i>left</i> , <i>right</i> , <i>center</i> , <i>justified</i>

tipo oggetto	attributo	valore
	weight	lo spessore del testo; valori possibili: <i>lighter</i> , <i>normal</i> , <i>bold</i> , <i>bolder</i>
	style	lo stile del testo; valori possibili: <i>normal</i> , <i>italic</i> , <i>oblique</i>
	height	altezza dell'area di scorrimento
	paused	true: ferma lo scorrimento; false: riparte
panel	visible	<i>true</i> : visibile; <i>false</i> : non visibile
	blink	<i>true</i> : lampeggio lento; <i>slow</i> : lampeggio lento; <i>fast</i> : lampeggio veloce; <i>false</i> : stop lampeggio
	pos	la posizione dell'oggetto
	color	<i>r</i> : rosso, <i>g</i> : verde, <i>b</i> : blu, <i>gr</i> : grigio, <i>y</i> : giallo. Impostare il colore a una stringa vuota equivale a impostare il colore di default del pannello (trasparente)
slider	visible	<i>true</i> : visibile; <i>false</i> : non visibile
	blink	<i>true</i> : lampeggio lento; <i>slow</i> : lampeggio lento; <i>fast</i> : lampeggio veloce; <i>false</i> : stop lampeggio
	value	<i>on</i> , <i>off</i> , un valore da 0% a 100%, da 0.00 a 1.00, da 0/<max> a <max>/<max>
submit	visible	<i>true</i> : visibile; <i>false</i> : non visibile
submitmini	blink	<i>true</i> : lampeggio lento; <i>slow</i> : lampeggio lento; <i>fast</i> : lampeggio veloce; <i>false</i> : stop lampeggio
submitmicro	pos	la posizione dell'oggetto
	text	il testo visualizzato
	color	<i>r</i> : rosso, <i>g</i> : verde, <i>b</i> : blu, <i>gr</i> : grigio, <i>y</i> : giallo
submitimage	visible	<i>true</i> : visibile; <i>false</i> : non visibile
	blink	<i>true</i> : lampeggio lento; <i>slow</i> : lampeggio lento; <i>fast</i> : lampeggio veloce; <i>false</i> : stop lampeggio
	pos	la posizione dell'oggetto
	img	nome del file immagine contenuto nella directory www/img
text	visible	<i>true</i> : visibile; <i>false</i> : non visibile
	blink	<i>true</i> : lampeggio lento; <i>slow</i> : lampeggio lento; <i>fast</i> : lampeggio veloce; <i>false</i> : stop lampeggio
	pos	la posizione dell'oggetto
	text	il testo visualizzato
	color	il colore del testo; utilizzare i codici di colore standard HTML
	size	la grandezza dei caratteri, in pixel

tipo oggetto	attributo	valore
	width	la larghezza del campo di testo, in pixel
	align	l'allineamento del testo; valori possibili: <i>left</i> , <i>right</i> , <i>center</i> , <i>justified</i>
	weight	lo spessore del testo; valori possibili: <i>lighter</i> , <i>normal</i> , <i>bold</i> , <i>bolder</i>
	style	lo stile del testo; valori possibili: <i>normal</i> , <i>italic</i> , <i>oblique</i>
user	visible	<i>true</i> : visibile; <i>false</i> : non visibile
usermini	blink	<i>true</i> : lampeggio lento; <i>slow</i> : lampeggio lento; <i>fast</i> : lampeggio veloce; <i>false</i> : stop lampeggio
usermicro	pos	la posizione dell'oggetto
	text	il testo visualizzato
	color	<i>r</i> : rosso, <i>g</i> : verde, <i>b</i> : blu, <i>gr</i> : grigio, <i>y</i> : giallo
userimage	visible	<i>true</i> : visibile; <i>false</i> : non visibile
	blink	<i>true</i> : lampeggio lento; <i>slow</i> : lampeggio lento; <i>fast</i> : lampeggio veloce; <i>false</i> : stop lampeggio
	pos	la posizione dell'oggetto
	img	nome del file immagine contenuto nella directory <code>www/img</code>
	text	il testo visualizzato
userrgb	visible	<i>true</i> : visibile; <i>false</i> : non visibile
	blink	<i>true</i> : lampeggio lento; <i>slow</i> : lampeggio lento; <i>fast</i> : lampeggio veloce; <i>false</i> : stop lampeggio
	pos	la posizione dell'oggetto
	text	il testo visualizzato
	color	il colore del tasto; utilizzare i codici di colore standard HTML
video	visible	<i>true</i> : visibile; <i>false</i> : non visibile
	blink	<i>true</i> : lampeggio lento; <i>slow</i> : lampeggio lento; <i>fast</i> : lampeggio veloce; <i>false</i> : stop lampeggio
	pos	la posizione dell'oggetto
	play/pause	riproduzione/pausa del video
	mode	modalità di riproduzione: <i>stop</i> , <i>auto</i> e <i>loop</i>

Esempi:

```
UISET mytext.style = italic
UISET mychart.barcolor = "#000000"
```


Programmazione Java

L'estensione Java di HSYCO è basata su una classe il cui nome deve essere obbligatoriamente **user**, dichiarata in un file **user.java**, contenuto nella directory **hsyco/com/hsyco** e compilato nel file **java.class**.

Il file **user.class** deve essere sempre presente per la corretta esecuzione di HSYCO, anche nel caso non siano state fatte personalizzazioni.

La classe *user* deve estendere la classe *userBase* di HSYCO, che contiene tutti i metodi di HSYCO per l'esecuzione di azioni sui sistemi di campo e lo scheletro di tutti i metodi di **callback**, cioè dei metodi richiamati da HSYCO in seguito agli eventi rilevati. Il file *user.java* non personalizzato, fornito con HSYCO insieme al corrispondente *user.class*, contiene solo la dichiarazione della classe:

```
/*  HSYCO CONTROLLER, USER CODE
 *   (c) 2012 Home Systems Consulting SpA
 *
 *   For information, see the Home Systems Consulting web site:
 *   http://www.homesystemsconsulting.com/
 */

package com.hsyco;

public class user extends userBase {
}
```

Per rendere operative le modifiche effettuate al file *user.class* è necessario riavviare HSYCO. L'impostazione standard del parametro **AutoKillFiles** in *hsyco.ini* prevede il riavvio automatico di HSYCO in caso di modifiche.

Nel seguito di questo capitolo vengono descritte dettagliatamente tutte le costanti, i metodi di callback e di comando implementati dalla classe *userBase*, e disponibili per la personalizzazione di *user.java*.

Costanti Predefinite

Nella tabella seguente sono elencate le costanti generiche utilizzabili come valore dei parametri dei metodi della classe *user*.

Name	Description
MERGED	DMX merged channel
NOCHANGE	don't change current status
OFF	action and status OFF
ON	action and status ON
PAUSE	status PAUSE (only Squeezebox music player)
PLAY	status PLAY (only Squeezebox music player)
UNKNOWN	unknown status value

Metodi Callback

I metodi descritti di seguito sono quelli invocati dal sistema di gestione eventi di HSYCO. Implementando questi metodi è possibile associare codice Java personalizzato a ciascun evento. In alcuni casi i valori di ritorno di questi metodi possono influenzare il comportamento di HSYCO.

Tutti i metodi propagano verso l'alto, cioè verso HSYCO, eventuali eccezioni. In questo caso HSYCO si occupa di generare un messaggio di errore nel log. Ad esempio:

```
OpenWebNetMonitor - Exception in user event call: ...
```

I/O Server

public static void IOStartupEvent(int serverIndex)

Evento che si verifica durante l'avvio dei thread di monitor di ciascuna interfaccia I/O, quindi all'inizio dell'esecuzione di HSYCO, ma anche in seguito ad ogni riavvio del thread di comunicazione tra HSYCO e l'interfaccia.

Parametri:

serverIndex - il numero progressivo dell'interfaccia, a partire da 0, basato sull'elenco definito nel parametro ioServers in hsyco.ini; se è presente un solo I/O Server, può essere omesso.

public static void IOEvent(String id, String value)

Richiamato al cambiamento di stato di ingresso o uscita di un I/O server.

Parametri:

id - identifica l'IO Server e lo specifico ingresso o uscita. Il formato dipende dal tipo di server gestito, ma generalmente è composto dal nome del server, come dichiarato nel parametro ioServers in hsyco.ini, seguito dal nome assegnato all'ingresso/uscita

value - il nuovo valore. Per ingressi o uscite binari, il valore restituito sarà "0" oppure "1".

Timer e Scheduler

public static boolean UserTimerEvent(String name, boolean active)

Questo metodo viene richiamato nel momento in cui un timer o la regola di uno scheduler viene attivata, e prima di eseguire le azioni associate. Viene inoltre eseguito alla disattivazione del timer o della regola dello scheduler. Questo metodo deve restituire il valore **true** perché il timer o la regola dello scheduler diventi effettivamente attiva e le azioni associate siano eseguite. Qualora il metodo restituisca **false**, il timer non sarà attivato, ma il metodo sarà richiamato ancora ogni minuto per tutta la durata prevista dal timer. Allo stesso modo, se il metodo restituisce **false** alla disattivazione del timer o della regola dello scheduler, questo non sarà disattivato, estendendo l'attivazione del timer oltre la durata impostata. Questo metodo è bloccante.

Parametri:

name - nome del timer o della regola dello scheduler

active - vale *true* all'attivazione e *false* alla disattivazione del timer o della regola dello scheduler.

DMX

public static void DmxStartupEvent(int serverIndex)

Eseguito durante l'avvio del thread di invio comandi di ciascun bus DMX, quindi una volta per bus all'inizio dell'esecuzione di HSYCO ma anche in seguito ad ogni riavvio del thread di invio comandi, ad esempio in seguito ad errori di comunicazione. Può essere utilizzato per inviare comandi DMX.

E' un metodo bloccante. L'esecuzione di questo metodo deve essere completata perché il corrispondente thread di comandi possa essere avviato.

Parametri:

serverIndex - il numero progressivo del bus DMX, a partire da 0.

public static void DmxEvent(int channel, int state)

Richiamato al cambiamento di stato sui bus DMX-512.

Parametri:

channel - indirizzo DMX, da 1 a 512 eventualmente preceduto dal numero progressivo di bus DMX se è presente più di un bus
state - il nuovo valore assunto dal canale, da 0 a 255.

public static int DmxFilter(int channel, int state, boolean reverse)

Questo metodo consente di filtrare il valore dello stato di ciascun canale sui bus DMX-512, inviando al gateway dei valori diversi da quelli impostati in HSYCO. In questo modo è possibile effettuare correzioni cromatiche, di intensità o posizione per ciascun canale.

Il metodo è bloccante e viene chiamato durante l'invio dei comandi al gateway DMX.

Viene anche chiamato in seguito alla lettura dello stato attuale del gateway. In questo caso la funzione di correzione applicata dovrebbe essere simmetrica e restituire un valore complementare.

Parametri:

- channel - indirizzo DMX, da 1 a 512 eventualmente preceduto dal numero progressivo di bus DMX se è presente più di un bus
- state - il valore non filtrato del canale, da 0 a 255
- reverse - *false* per le chiamate in scrittura verso il gateway; *true* per le chiamate in lettura dal gateway.

Telecamere

public static void CameraMotionEvent(String eventName, long remoteTime)

Richiamato in seguito ad un evento di motion detection notificato da una telecamera inviando al Web Server di HSYCO una richiesta tipo:

```
http://192.168.0.50/x/camerarec?camera=<name>&zone=<id>
```

Il separatore tra i parametri *camera* e *zone* può essere “.” oppure “;”.

La registrazione può essere attivata anche tramite il metodo Java CameraRecTrigger() o le azioni CAMERAREC o CAMERARECFULL in EVENTS.

Parametri:

eventName - l'identificativo dell'evento, nel formato<name>:<id>

remoteTime - il tempo in millisecondi

public static int cameraCommandEvent(String function, String action, String camera)


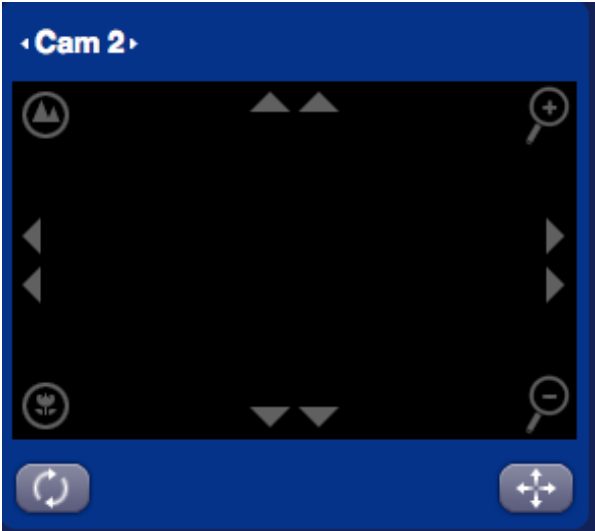





Richiamato in seguito ad un comando PTZ richiesto dall'interfaccia Web di HSYCO su una telecamera per la quale sia definito qualunque driver PTZ, compreso *user*.

All'interno di questo metodo è possibile inserire il codice Java necessario per eseguire movimenti PTZ e messa a fuoco per telecamere non supportate da HSYCO, oppure sostituire l'esecuzione di alcune delle funzioni PTZ standard con azioni personalizzate.

Se questa funzione ritorna un valore diverso da -1, allora HSYCO eseguirà il comando PTZ previsto per la telecamera. Restituendo -1 il comando PTZ non sarà eseguito.

Parametri:

function - può assumere i valori definiti nella tabella seguente
action - può assumere i valori definiti nella tabella seguente
camera - il nome della telecamera.

	function	action	
	focus	far	
	focus	near	
	zoom	tele	
	zoom	wide	
	move	left	
	move	right	
	move	up	
	move	down	
Center	move	stop	

IRTrans

public static void IREvent(boolean received, int irtransid, String event)

Questo metodo viene eseguito quando uno degli IRTrans riceve o emette un comando IR presente nel database dei comandi memorizzati.

Parametri:

received - *true* se il comando è stato ricevuto²⁰ dall'IRTrans, *false* se è stato emesso

irtransid - il numero dell'IRTrans, a partire da 0

event - la stringa del comando ricevuto o emesso nel formato *<telecomando>,<comando>*.

²⁰ Cioè se il dispositivo ha rilevato un comando emesso da un normale telecomando a infrarossi.

Squeezebox

public static void SlimPowerEvent(int index, int power)

Richiamato all'accensione o spegnimento degli Squeezebox.

Parametri:

index - il numero dello Squeezebox, a partire da 0

power - ON o OFF.

public static void SlimStatusEvent(int index, int status)

Richiamato al cambiamento di stato degli Squeezebox.

Parametri:

index - il numero dello Squeezebox, a partire da 0

status - ON, OFF, PLAY, PAUSE, UNKNOWN.

public static void SlimVolumeEvent(int index, int volume)

Richiamato al cambiamento di volume degli Squeezebox.

Parametri:

index - il numero dello Squeezebox, a partire da 0

volume - un numero intero che indica la variazione²¹ del volume in una scala da 0 a 100; numeri positivi indicano un aumento del volume, numeri negativi una riduzione.

²¹ Non viene restituito il valore assoluto del volume, ma l'entità della variazione richiesta dal comando che ha generato l'evento

PBX

public static boolean PBXCallEvent(String host, String caller, String called)

Richiamato a seguito di ogni notifica di chiamata dalla centrale telefonica interfacciata a HSYCO. Se il metodo restituisce *true*, la richiesta viene registrata sul file di log con stato [OK], altrimenti con [ERROR].

Parametri:

host - l'indirizzo IP del server PBX

caller - il numero o identificativo chiamante

called - il numero chiamato.

Servizi di Rete

public static void LocationEvent(long MAC, InetAddress IP, int zoneld)

Questo metodo viene richiamato, se il servizio di localizzazione è attivo, quando viene rilevata una variazione nell'associazione di un client rispetto agli Access Point Wi-Fi.

Parametri:

MAC - il MAC address del client

IP - l'indirizzo IP del client

zoneld - in caso di associazione è il numero progressivo a partire da 0 dell'Access Point, in base all'ordine definito nel parametro *LocationBases* in hsyco.ini; -1 se è stata rilevata una de-associazione.

Metodi di Sistema

public static void StartupEvent()

Metodo eseguito una sola volta all'avvio di HSYCO, subito dopo l'inizializzazione, ma prima dell'esecuzione dei vari thread di monitor e del Web Server.

Non viene eseguito in un thread, quindi è bloccante, nel senso che questo metodo deve completare la sua esecuzione prima che i vari servizi di HSYCO possano essere avviati.

public static String getUserVersion()

Questo metodo restituisce una stringa, utilizzata per la generazione della riga di log all'avvio di HSYCO. Si raccomanda di restituire una stringa che specifichi la versione di user.java.

public static void TimeEvent(long time)

Richiamato ogni 60 secondi. HSYCO cerca di sincronizzare l'esecuzione all'inizio del minuto, cioè di eseguire questo metodo al secondo 00 di ogni minuto. Tuttavia l'esecuzione non è garantita. In condizioni di carico eccezionale HSYCO potrebbe non essere in grado di eseguire la chiamata in modo regolare.

Parametri:

time - il tempo attuale in millisecondi.

public static void programTimerEvent(String name)

Richiamato in seguito all'attivazione di un program timer.

I program timer vengono impostati con i metodi `programTimerSet()`, `programTimerReset()`, `programTimerRepeat()`, oppure utilizzando le corrispondenti azioni in EVENTS.

Parametri:

name - nome del program timer.

public static String userCommand(String name, String param)

Chiamato in seguito alla pressione dei tasti creati nell'interfaccia Web con gli oggetti (user), (usermini), (usermicro), (userimage) o (userrgb), oppure quando in EVENTS viene eseguita l'azione USER name=param.

Viene anche chiamato in seguito alla pressione degli oggetti (submit!id), (submitmini!id) or (submitmicro!id), utilizzati per la gestione dei form.

Questo metodo restituisce una stringa. Se il valore restituito è null, la chiamata viene tracciata nel file di log con un flag [ERROR], altrimenti con [OK]. Se si vuole evitare la scrittura nel file di log dei tasti (user) premuti, è sufficiente che questo metodo restituisca una stringa che inizia con il carattere "!".

Infine, per visualizzare una nuova pagina a seguito della pressione del tasto (user) o (submit), restituire una stringa che inizia con "page:" seguito dal nome della pagina da visualizzare; in questo caso, la chiamata del metodo `userCommand()` verrà eseguita di nuovo quando il pop-up o la pagina verranno chiusi, con l'aggiunta finale di "/close" a param.

Parametri:

name - corrisponde al campo <name> dell'oggetto (user). Se richiamato dall'oggetto (submit!id), corrisponde all'id dell'oggetto

param - corrisponde al campo <param> dell'oggetto (user). Se richiamato dall'oggetto (submit!id), viene impostato con le coppie id-valore degli oggetti (input) presenti nella pagina o nel container, separati dal carattere "@".

public static void DaylightEvent(boolean day)

Richiamato all'avvio di HSYCO e successivamente ad ogni transizione giorno-notte, in base ai valori di latitudine e longitudine impostati in hsyco.ini.

Parametri:

day - *true* durante il giorno, *false* durante la notte.

public static int PowerEvent(int power)

Richiamato a seguito di un cambiamento del livello di potenza, impostato utilizzando il metodo Java powerSet() o l'azione POWER.

Se PowerEvent() restituisce -1 o non è presente in user.java, lo stato di HSYCO viene aggiornato con il valore di potenza rilevato, altrimenti viene salvato il valore restituito da PowerEvent(). Grazie a questo metodo è quindi possibile alterare il valore di potenza visualizzato nell'interfaccia Web rispetto al valore misurato, ad esempio per aggregare letture di potenza acquisite da più sensori.

Parametri:

power - la potenza espressa in Watt.

public static void SunPositionEvent(int azimuth, int elevation)

Richiamato ad ogni variazione dell'angolo di elevazione o di azimuth del sole rispetto alla posizione impostata in hsyco.ini.

Parametri:

azimuth - l'angolo che il sole forma con il nord vero, in gradi decimali interi

elevation - l'angolo che il sole forma con la linea dell'orizzonte, in gradi decimali interi, negativi dopo il tramonto.

public static String WebRootRequestEvent(InetAddress addr, boolean secure, String useragent)

Richiamato quando il Web Server riceve una richiesta della home page del server. E' possibile restituire una stringa per forzare il redirect del browser ad un URL valido, oppure null per prevenire la re-direzione.

Parametri:

addr - indirizzo del browser da cui è arrivata la richiesta
secure - true se la richiesta è stata fatta con il protocollo HTTPS
useragent - informazioni sul browser che ha generato la richiesta.

static void SchedulerEvent(String groupname, String schedulename)

Questo metodo di tipo callback bloccante viene richiamato ad intervalli di tempo definiti dal metodo SchedulerRegister().

Ad ogni scheduler viene associato un nome di gruppo ed un nome proprio dello scheduler. Gli scheduler che appartengono allo stesso gruppo vengono eseguiti sequenzialmente, quindi uno alla volta, mentre scheduler di gruppi diversi possono essere eseguiti in parallelo.

Parametri:

groupname - nome del gruppo
schedulename - nome proprio dello scheduler.

Metodi di Comando e Utilità

I metodi descritti di seguito consentono di eseguire comandi rivolti ai dispositivi di campo, di acquisire informazioni sullo stato dei dispositivi stessi e di avere accesso ad alcune funzioni di utilità generale.

Per tutti i metodi che eseguono azioni o inviano comandi, viene scritta una riga nel file di log del tipo:

2011.07.30 11:39:16.133 - USERCODE: <metodo> - <parametri>

I/O Server

static void ioSet(String id, String value)

Imposta variabili di I/O Server o data point.

Parametri:

id - id del server e nome della porta di ingresso o di uscita. Il formato dipende dal tipo di server gestito, ma generalmente è composto dal nome del server, come dichiarato nel parametro ioServers in hsyco.ini, seguito dal nome assegnato all'ingresso/uscita, ad esempio "contacts.o1"

value - il nuovo valore.

static String ioGet(String id)

Restituisce lo stato corrente di un ingresso o uscita di un gateway I/O.

Parametri:

id - id del server e nome della porta di ingresso o di uscita. Il formato dipende dal tipo di server gestito, ma generalmente è composto dal nome del server, come dichiarato nel parametro ioServers in hsyco.ini.

DMX

static int dmxCSet(int channel, int state)

Invia un valore ad un canale sul bus DMX-512.

Restituisce 1 in caso di successo, 0 in caso di errore.

Parametri:

channel - indirizzo DMX, da 1 a 512 eventualmente preceduto dal numero progressivo di bus DMX se è presente più di un bus

state - il nuovo valore assunto dal canale, da 0 a 255.

static int dmxCSet(int[] channels, int[] state)

Invia una lista di nuovi valori ad un gruppo di canali sul bus DMX-512. Questo metodo viene utilizzato in alternativa all'impostazione di un indirizzo singolo per migliorare le prestazioni e garantire un cambiamento di stato rapido quando devono essere impostati molti dispositivi contemporaneamente.

Quando viene eseguito questo metodo, così come e i metodi dmxCOn() e dmxCOff(), la variazione anche di uno solo dei canali indicati nell'array provoca l'esecuzione dei metodi callback associati agli eventi DMX per tutti i canali presenti nell'array, anche quelli non modificati da questa chiamata.

Restituisce 1 in caso di successo, 0 in caso di errore.

Parametri:

channels - array di indirizzi DMX, da 1 a 512 eventualmente preceduti dal numero progressivo di bus DMX se è presente più di un bus

state - il nuovo valore assunto dal canale, da 0 a 255.

static int dmxCff(int channel)

Imposta un canale sul bus DMX-512 al valore 0.

Restituisce 1 in caso di successo, 0 in caso di errore.

Parametri:

channel - indirizzo DMX, da 1 a 512 eventualmente preceduto dal numero progressivo di bus DMX se è presente più di un bus.

static int dmxCff(int[] channels)

Imposta un gruppo di canali sul bus DMX-512 al valore 0. Questo metodo viene utilizzato in alternativa all'impostazione di un indirizzo singolo per migliorare le prestazioni e garantire un cambiamento di stato rapido quando devono essere impostati molti dispositivi contemporaneamente.

Quando viene eseguito questo metodo, così come e i metodi dmxCn() e dmxCff(), la variazione anche di uno solo dei canali indicati nell'array provoca l'esecuzione dei metodi callback associati agli eventi DMX per tutti i canali presenti nell'array, anche quelli non modificati da questa chiamata.

Restituisce 1 in caso di successo, 0 in caso di errore.

Parametri:

channels - array di indirizzi DMX, da 1 a 512 eventualmente preceduti dal numero progressivo di bus DMX se è presente più di un bus
state - il nuovo valore assunto dal canale, da 0 a 255.

static int dmxCn(int channel)

Imposta un canale sul bus DMX-512 al valore precedente all'ultima chiamata a dmxCff().

Restituisce 1 in caso di successo, 0 in caso di errore.

Parametri:

channel - indirizzo DMX, da 1 a 512 eventualmente preceduto dal numero progressivo di bus DMX se è presente più di un bus.

static int dmxOn(int[] channels)

Imposta un gruppo di canali sul bus DMX-512 ai valori precedenti all'ultima chiamata a dmxOff() per ciascun canale. Questo metodo viene utilizzato in alternativa all'impostazione di un indirizzo singolo per migliorare le prestazioni e garantire un cambiamento di stato rapido quando devono essere impostati molti dispositivi contemporaneamente.

Restituisce 1 in caso di successo, 0 in caso di errore.

Parametri:

channels - array di indirizzi DMX, da 1 a 512 eventualmente preceduti dal numero progressivo di bus DMX se è presente più di un bus.

static int dmxMerge(int from, int to, boolean merge)

Imposta i canali DMX compresi tra i parametri *from* e *to* in modalità **MERGE** oppure ritorna al funzionamento normale. In modo MERGE i canali sul bus DMX OUT del gateway seguono i valori dei canali del bus DMX IN.

Restituisce 1 in caso di successo, -1 se *to* < *from*, 0 in caso di errore generico.

Parametri:

from - indirizzo DMX, da 1 a 512 eventualmente preceduto dal numero progressivo di bus DMX se è presente più di un bus

to - indirizzo DMX, da 1 a 512 eventualmente preceduto dal numero progressivo di bus DMX se è presente più di un bus

merge - *true* per impostare la modalità MERGE, *false* per ritornare al modo di funzionamento normale.

static int dmxFMerge(int[] channels, boolean merge)

Imposta un gruppo di canali sul bus DMX-512 in modalità **MERGE** oppure ritorna al funzionamento normale.

In modo MERGE i canali sul bus DMX OUT del gateway seguono i valori dei canali del bus DMX IN.

Restituisce 1 in caso di successo, 0 in caso di errore.

Parametri:

channels - array di indirizzi DMX, da 1 a 512 eventualmente preceduti dal numero progressivo di bus DMX se è presente più di un bus.

static int dmxFGet(int channel)

Restituisce lo stato corrente di un canale sul bus DMX-512, con valori compresi tra 0 e 255, oppure la costante predefinita UNKNOWN se lo stato è sconosciuto.

Parametri:

channel - indirizzo DMX, da 1 a 512 eventualmente preceduto dal numero progressivo di bus DMX se è presente più di un bus.

static void dmxFOffTimerSet(int channel, int seconds)

Questo metodo consente di impostare un comando di spegnimento a tempo per un canale DMX-512.

Se il dispositivo è in stato ON (valore del canale maggiore di 0) al momento dell'esecuzione di questo metodo, HSYCO si occuperà di impostare il canale al valore 0 esattamente dopo il numero di secondi specificato dal parametro *seconds*.

Questo metodo viene ignorato se è già impostato un altro timer per lo stesso canale. Inoltre un timer impostato con questo metodo viene cancellato automaticamente se il canale cambia stato. Di conseguenza non ha senso utilizzare questo metodo per impostare un timer su un canale il cui valore corrente sia 0.

Parametri:

channel - indirizzo DMX, da 1 a 512 eventualmente preceduto dal numero progressivo di bus DMX se è presente più di un bus
seconds - il tempo in secondi, a partire dall'istante di esecuzione del metodo, che deve trascorrere prima che HSYCO invii automaticamente il comando OFF al dispositivo.

static void dmxClearTimer(int channel)

Cancella un timer di spegnimento precedentemente impostato.

Parametri:

channel - indirizzo DMX, da 1 a 512 eventualmente preceduto dal numero progressivo di bus DMX se è presente più di un bus.

static void dmxClearTimerPreset(int channel, int seconds)

Questo metodo è simile a *dmxClearTimerSet()*, con la differenza di poter essere eseguito quando il canale è in stato OFF. Nel momento in cui un altro comando sul bus porterà il canale in stato ON, il timer pre-impostato diventerà attivo, provocando lo spegnimento dopo il numero di secondi impostato.

Il metodo viene ignorato se è già impostato un altro timer per lo stesso canale.

Parametri:

channel - indirizzo DMX, da 1 a 512 eventualmente preceduto dal numero progressivo di bus DMX se è presente più di un bus
seconds - il tempo in secondi, a partire dall'istante in cui il dispositivo passa in stato ON, che deve trascorrere prima che HSYCO invii automaticamente il comando OFF.

static void dmxOffTimerReset(int channel, int seconds)

Questo metodo consente di impostare un comando di spegnimento a tempo per un canale DMX-512.

Diversamente da *dmxOffTimerSet()*, qualora sia già presente un timer per lo stesso canale, il timer precedente verrà eliminato e sostituito con quello impostato con questo metodo.

Parametri:

channel - indirizzo DMX, da 1 a 512 eventualmente preceduto dal numero progressivo di bus DMX se è presente più di un bus
seconds - il tempo in secondi, a partire dall'istante di esecuzione del metodo oppure a partire dall'istante in cui il dispositivo passa in stato ON, che deve trascorrere prima che HSYCO invii automaticamente il comando OFF.

Telecamere

static void cameraRecTrigger(String camera, String source, int seconds)

Attiva la registrazione dei frame acquisiti da una telecamera.

Parametri:

- camera - il nome della telecamera, compreso nell'elenco definito dal parametro Cameras in hsyco.ini
- source - una stringa utilizzata a livello informativo per generare il messaggio di log
- seconds - la durata della registrazione in secondi. Se impostato a 0 termina la registrazione in corso.

static void cameraRecTriggerFull(String camera, String source, int seconds)

Identico al metodo cameraRecTrigger(), ma con questo comando viene ignorata l'opzione DroppedFrames eventualmente specificata in hsyco.ini, registrando quindi tutti i frame per il periodo di tempo indicato.

Parametri:

- camera - il nome della telecamera, compreso nell'elenco definito dal parametro Cameras in hsyco.ini
- source - una stringa utilizzata a livello informativo per generare il messaggio di log
- seconds - la durata della registrazione in secondi. Se impostato a 0 termina la registrazione in corso.

static int cameraCommand(String function, String action, String camera)








Consente di inviare comandi PTZ e messa a fuoco alle telecamere abilitate. Restituisce un valore utilizzato solamente nei messaggi di log.

Parametri:

function - può assumere i valori definiti nella tabella

action - può assumere i valori definiti nella tabella

camera - il nome della telecamera.

	function	action
	focus	far
	focus	near
	zoom	tele
	zoom	wide
	move	left
	move	right
	move	up
	move	down
Centro	move	stop

static void cameraMode(String camera, boolean enabled)

Consente di abilitare o disabilitare temporaneamente una telecamera. Quando la telecamera è disabilitata vengono disattivate anche le funzioni di registrazione.

Parametri:

camera - il nome della telecamera

enabled - *true* per abilitare la telecamera, *false* per disabilitarla.

static void cameraRecMode(String camera, boolean enabled)

Consente di abilitare o disabilitare temporaneamente le funzioni di registrazione di una telecamera, mantenendo comunque attiva la telecamera per l'accesso alle immagini in tempo reale.

Parametri:

camera - il nome della telecamera

enabled - *true* per abilitare la registrazione, *false* per disabilitarla.

Porte di Comunicazione Seriale

static int writeComm(String portName, String data)

Invia una sequenza di byte alla porta seriale specificata.

Restituisce 0 in caso di errore, oppure il numero di byte inviati in caso di successo.

Se `verboseLog = true` e `userLog = true`, i byte ricevuti e trasmessi sulla porta vengono trascritti nel file di log.

Parametri:

`portName` - il nome della porta di comunicazione, come definito dal parametro **CommPorts** in `hsyco.ini`

`data` - una stringa con la rappresentazione esadecimale della sequenza di byte da inviare. Ad esempio "45AB4400BA".

static String readComm(String portName, int len)

Legge una sequenza di byte dalla porta di comunicazione specificata. Il numero di byte letti dipende dalla disponibilità dei dati nel buffer della porta di comunicazione²², fino al numero massimo specificato nel parametro `len`.

Se `len` viene impostato a 0, la funzione non restituisce dati, ma svuota il buffer della porta di comunicazione dai dati eventualmente presenti al momento della chiamata.

Restituisce `null` in caso di errore, una stringa con la rappresentazione esadecimale della sequenza di byte ricevuti o una stringa vuota nel caso in cui il buffer dei dati in ingresso sia vuoto. Ad esempio "45AB4400BA".

²² E' previsto un time-out di 2 secondi sulla lettura dei dati. Nel caso in cui il buffer sia vuoto al momento dell'esecuzione della funzione, questa rimane in attesa fino al time-out.

Se `verboseLog = true` e `userLog = true`, i byte ricevuti e trasmessi sulla porta vengono trascritti nel file di log.

Parametri:

`portName` - il nome della porta di comunicazione, come definito dal parametro ***CommPorts*** in `hsyco.ini`

`len` - numero massimo di byte da leggere dalla porta, oppure 0 per svuotare il buffer senza leggere i dati.

static int closeComm(String portName)

Chiude la connessione verso la porta seriale quando la porta seriale è dichiarata di tipo “server”, altrimenti viene ignorato.

Quando viene utilizzato con server seriali in configurazione di fail-over, questo metodo forza una connessione alla prossima chiamata del metodo `readComm()` o `WriteComm()`, in modo tale che ci sia uno scambio tra gli IP in fail-over se l'IP corrente non risponde.

Parametri:

`portName` - il nome della porta di comunicazione, come definito dal parametro ***CommPorts*** in `hsyco.ini`

Plugins

static Bentel getBentelPlugIn(String id)

Consente di utilizzare in user.java i metodi della classe add-on di integrazione dei sistemi di sicurezza Bentel. L'utilizzo di questo add-on è descritto nel documento Bentel Application Note.

Restituisce l'istanza della classe Bentel associata all'id, o null in caso di errore.

Parametri:

id - l'id del sistema Bentel, come impostato nel file di configurazione bentel.ini.

static ParadoxEVO getParadoxEVOPlugIn(String id)

Consente di utilizzare in user.java i metodi della classe add-on di integrazione dei sistemi di sicurezza Paradox EVO. L'utilizzo di questo add-on è descritto nel documento Paradox EVO Application Note.

Restituisce l'istanza della classe ParadoxEVO associata all'id, o null in caso di errore.

Parametri:

id - l'id del sistema EVO, come impostato nel file di configurazione paradox.ini.

static Tecnoalarm getTecnoalarmPlugIn(String id)

Consente di utilizzare in user.java i metodi della classe add-on di integrazione dei sistemi di sicurezza Tecnoalarm. L'utilizzo di questo add-on è descritto nel documento Tecnoalarm Application Note.

Restituisce l'istanza della classe Tecnoalarm associata all'id, o null in caso di errore.

Parametri:

id - l'id del sistema Tecnoalarm, come impostato nel file di configurazione bentel.ini.

Modbus

static byte[] modbusReadCoils(String name, int unit, int address, int quantity)

Legge il contenuto di un blocco contiguo di coils da un dispositivo MODBUS, utilizzando il codice funzione 0x01.

Restituisce una PDU di risposta standard. In caso di risposta corretta, il primo byte assume il valore 0x02, il secondo indica il numero di byte del blocco dati e i dati letti iniziano dal terzo byte. In caso di errore, il primo byte assume il valore 0x81 ed il secondo contiene il codice di errore.

Questo metodo può essere chiamato passando come nome del parametro un server I/O predefinito, il nome dell'host, o l'indirizzo IP del gateway MODBUS. In questo modo è possibile chiamare il metodo senza definire un server I/O per il gateway.

Parametri:

- name - l'id dell'I/O Server come definito in hsyco.ini, oppure il nome o l'indirizzo IP del gateway
- unit - l'indirizzo del dispositivo MODBUS slave (da 1 a 247)
- address - l'indirizzo del primo blocco di coils
- quantity - la quantità di coils da leggere (da 1 a 2000).

static byte[] modbusReadDiscreteInputs(String name, int unit, int address, int quantity)

Legge il contenuto di un blocco contiguo di discrete inputs da un dispositivo MODBUS, utilizzando il codice funzione 0x02.

Restituisce una PDU di risposta standard. In caso di risposta corretta, il primo byte assume il valore 0x02, il secondo indica il numero di byte del blocco dati e i

dati letti iniziano dal terzo byte. In caso di errore, il primo byte assume il valore 0x82 ed il secondo contiene il codice di errore.

Questo metodo può essere chiamato passando come nome del parametro un server I/O predefinito, il nome dell'host, o l'indirizzo IP del gateway MODBUS. In questo modo è possibile chiamare il metodo senza definire un server I/O per il gateway.

Parametri:

- name - l'id dell'I/O Server come definito in hsyco.ini, oppure il nome o l'indirizzo IP del gateway
- unit - l'indirizzo del dispositivo MODBUS slave (da 1 a 247)
- address - l'indirizzo del primo blocco di input discreti
- quantity - la quantità di input discreti da leggere (da 1 a 2000).

static byte[] modbusReadHoldingRegisters(String name, int unit, int address, int quantity)

Legge il contenuto di un blocco contiguo di holding register da un dispositivo MODBUS, utilizzando il codice funzione 0x03.

Restituisce una PDU di risposta standard. In caso di risposta corretta, il primo byte assume il valore 0x03, il secondo indica il numero di byte del blocco dati e i dati letti iniziano dal terzo byte. In caso di errore, il primo byte assume il valore 0x83 ed il secondo contiene il codice di errore.

Questo metodo può essere chiamato passando come nome del parametro un server I/O predefinito, il nome dell'host, o l'indirizzo IP del gateway MODBUS. In questo modo è possibile chiamare il metodo senza definire un server I/O per il gateway.

Parametri:

- name - l'id dell'I/O Server come definito in hsyco.ini, oppure il nome o l'indirizzo IP del gateway
- unit - l'indirizzo del dispositivo MODBUS slave (da 1 a 247)

address - l'indirizzo del primo registro del blocco da leggere

quantity - la quantità di registri da leggere (da 1 a 125).

static byte[] modbusReadInputRegisters(String name, int unit, int address, int quantity)

Legge il contenuto di un blocco contiguo di input register da un dispositivo MODBUS, utilizzando il codice funzione 0x04.

Restituisce una PDU di risposta standard. In caso di risposta corretta, il primo byte assume il valore 0x04, il secondo indica il numero di byte del blocco dati e i dati letti iniziano dal terzo byte. In caso di errore, il primo byte assume il valore 0x84 ed il secondo contiene il codice di errore.

Questo metodo può essere chiamato passando come nome del parametro un server I/O predefinito, il nome dell'host, o l'indirizzo IP del gateway MODBUS. In questo modo è possibile chiamare il metodo senza definire un server I/O per il gateway.

Parametri:

name - l'id dell'I/O Server come definito in hsyco.ini, oppure il nome o l'indirizzo IP del gateway

unit - l'indirizzo del dispositivo MODBUS slave (da 1 a 247)

address - l'indirizzo del primo registro del blocco da leggere

quantity - la quantità di registri da leggere (da 1 a 125).

static byte[] modbusWriteSingleCoil(String name, int unit, int address, boolean value)

Scrive un singolo coil in un dispositivo MODBUS, utilizzando il codice funzione 0x05.

Restituisce una PDU di risposta standard. In caso di risposta corretta, il primo byte assume il valore 0x05 and e gli ultimi due byte 0xFF00 se il coil è on,

0x0000 se il coil è off. In caso di errore, il primo byte assume il valore 0x85 ed il secondo contiene il codice di errore.

Questo metodo può essere chiamato passando come nome del parametro un server I/O predefinito, il nome dell'host, o l'indirizzo IP del gateway MODBUS. In questo modo è possibile chiamare il metodo senza definire un server I/O per il gateway.

Parametri:

- name - l'id dell'I/O Server come definito in hsyco.ini, oppure il nome o l'indirizzo IP del gateway
- unit - l'indirizzo del dispositivo MODBUS slave (da 1 a 247)
- address - l'indirizzo del coil
- value - vero per attivare il coil, falso per disattivarlo.

static byte[] modbusWriteSingleRegister(String name, int unit, int address, byte[] bytes)

Scrive un singolo registro in un dispositivo MODBUS, utilizzando il codice funzione 0x06.

Restituisce una PDU di risposta standard. In caso di risposta corretta, il primo byte assume il valore 0x06 e gli ultimi due byte il valore del registro. In caso di errore, il primo byte assume il valore 0x86 ed il secondo contiene il codice di errore.

Questo metodo può essere chiamato passando come nome del parametro un server I/O predefinito, il nome dell'host, o l'indirizzo IP del gateway MODBUS. In questo modo è possibile chiamare il metodo senza definire un server I/O per il gateway.

Parametri:

- name - l'id dell'I/O Server come definito in hsyco.ini, oppure il nome o l'indirizzo IP del gateway

unit - l'indirizzo del dispositivo MODBUS slave (da 1 a 247)

address - l'indirizzo del registro

bytes - i dati da scrivere nei registri, in formato big-endian (il byte più significativo è inviato per primo).

static byte[] modbusWriteMultipleRegisters(String name, int unit, int address, byte[] bytes)

Scrive un blocco contiguo di input register in un dispositivo MODBUS, utilizzando il codice funzione 0x10.

Restituisce una PDU di risposta standard. In caso di risposta corretta, il primo byte assume il valore 0x10, i due byte successivi indicano l'indirizzo del primo registro scritto e gli ultimi due byte la quantità di registri scritti. In caso di errore, il primo byte assume il valore 0x90 ed il secondo contiene il codice di errore.

Questo metodo può essere chiamato passando come nome del parametro un server I/O predefinito, il nome dell'host, o l'indirizzo IP del gateway MODBUS. In questo modo è possibile chiamare il metodo senza definire un server I/O per il gateway.

Parametri:

name - l'id dell'I/O Server come definito in hsyco.ini, oppure il nome o l'indirizzo IP del gateway

unit - l'indirizzo del dispositivo MODBUS slave (da 1 a 247)

address - l'indirizzo del primo registro del blocco da scrivere

bytes - i dati da scrivere nei registri, in formato big-endian (il byte più significativo è inviato per primo).

static byte[] modbusWriteMaskRegister(String name, int unit, int address, byte[] mask, byte[] bytes)

Scrive un registro con una maschera di bit register in un dispositivo MODBUS, utilizzando il codice funzione 0x16.

Restituisce una PDU di risposta standard. In caso di risposta corretta, il primo byte assume il valore 0x16, i successivi due byte il valore della mask e gli ultimi due il valore del registro. In caso di errore, il primo byte assume il valore 0x96 ed il secondo contiene il codice di errore.

Questo metodo può essere chiamato passando come nome del parametro un server I/O predefinito, il nome dell'host, o l'indirizzo IP del gateway MODBUS. In questo modo è possibile chiamare il metodo senza definire un server I/O per il gateway.

Parametri:

- name - l'id dell'I/O Server come definito in hsyco.ini, oppure il nome o l'indirizzo IP del gateway
- unit - l'indirizzo del dispositivo MODBUS slave (da 1 a 247)
- address - l'indirizzo del registro
- mask - la AND mask da applicare al registro
- bytes - i dati da scrivere nei registri, in formato big-endian (il byte più significativo è inviato per primo).

IRTrans

static int irtransCommand(String irlId, String command)

Consente di inviare un comando ad un IRTrans. Restituisce 0 in caso di errore.

Parametri:

irlId - identifica l'IRTrans e deve corrispondere ad uno dei nomi presenti nella lista del parametro IRTrans in hsyco.ini

command - la sequenza di comandi IR da inviare. Ogni comando è costituito dal nome che identifica il database di comandi²³, seguito da virgola o punto e dal nome del comando all'interno del database. Più comandi in sequenza possono essere separati da “;”. E' possibile inserire il comando speciale @N, che introduce un tempo di attesa pari a N millisecondi prima di inviare il comando successivo. Ad esempio:

```
panasonic,onoff;denon,on;@4000;denon,tv
```

invia il comando onoff del telecomando panasonic, seguito dal comando on del telecomando denon ed infine dal comando tv sempre del denon, inviando quest'ultimo comando dopo 4 secondi di pausa.

Inoltre il comando speciale @0 consente di inviare comandi IR senza tempi di attesa prima dell'invio dei comandi successivi²⁴. Ad esempio:

```
denon,on;@0;nad,on
```

invia il comando on del telecomando denon, seguito immediatamente dal comando on del telecomando nad.

²³ Ogni IRTrans può contenere un database di diversi telecomandi, e per ogni telecomando vengono memorizzati i diversi comandi possibili. Se nella directory hsyco/ir è presente un file .ccfhex corrispondente al nome del database, questo metodo userà il codice CCF presente nel file, dandogli precedenza rispetto al comando memorizzato nell'IRTrans stesso.

²⁴ Anche se non è specificato un ritardo @N prima di un comando IR, questo metodo prevede comunque una pausa di 500ms prima dell'invio di ciascun comando; con @0 questa pausa viene ridotta ad un tempo di 1ms per i comandi successivi.

Squeezebox

static String slimCommand(int index, String command)

Consente di inviare qualunque comando ad uno Squeezebox. Restituisce *null* in caso di errore, altrimenti il messaggio restituito in seguito all'esecuzione del comando.

Parametri:

index - il numero dello Squeezebox, a partire da 0, nell'ordine definito dal parametro ***slimPlayers*** in hsyco.ini
command - il comando da inviare²⁵.

static String slimCommand(String playerName, String command)

Consente di inviare qualunque comando ad uno Squeezebox. Restituisce *null* in caso di errore, altrimenti il messaggio restituito in seguito all'esecuzione del comando.

Parametri:

playerName - il nome dello Squeezebox, corrispondente a uno dei nomi elencati dal parametro ***slimPlayers*** in hsyco.ini
command - il comando da inviare.

static int slimButton(int index, String button)

Consente di inviare comandi agli Squeezebox. Restituisce 0 in caso di errore, 1 in caso di successo.

²⁵ I comandi sono definiti nella documentazione della Command Line Interface (CLI) di SqueezeCenter.

Parametri:

index - il numero dello Squeezebox, a partire da 0, nell'ordine definito dal parametro ***slimPlayers*** in hsyco.ini

button - i comandi elencati nella tabella seguente, oppure qualunque comando previsto dall'istruzione ***button*** della Command Line Interface di SqueezeCenter. E' possibile specificare più comandi separati dal carattere “;”.

button	Descrizione
x_volupxN	incrementa il volume del valore N
x_voldnxN	decrementa il volume del valore N
x_poweron	accende lo Squeezebox
x_poweroff	spegne lo Squeezebox
x_pauseon	pausa
x_pauseoff	esce dalla pausa e passa al modo play se possibile
unsync	elimina la sincronizzazione di questo player con un altro e lo spegne
syncN	sincronizza il player N con questo player; accende entrambi i player

static int slimButton(String playerName, String button)

Consente di inviare comandi agli Squeezebox. Restituisce 0 in caso di errore, 1 in caso di successo.

Parametri:

playerName - il nome dello Squeezebox, corrispondente a uno dei nomi elencati dal parametro ***slimPlayers*** in hsyco.ini

button - descritto nella tabella precedente.

Public Announcement

I metodi `audioPlay()` sono utilizzati per riprodurre messaggi vocali o file audio registrati.

Gli annunci vocali possono essere inviati al browser Web, oppure tramite l'uscita audio line-out del server, attraverso l'altoparlante interno o l'uscita audio delle telecamere Axis e l'altoparlante dei telefoni SNOM o dei dispositivi PA. Per maggiori dettagli consultare l'Appendice Public Announcement.

Destinazione audio	Descrizione
speaker	uscita audio line-out del server
web	uscita audio del browser Web (funzione supportata solo su Firefox e Chrome)
axis@<camera id>	messaggio vocale inviato a una telecamera Axis. Impostare camera id in base al nome definito per il parametro Cameras in hsyco.ini
snom@<ip:port>	messaggio vocale inviato ai telefoni SNOM e ai dispositivi di public announcement, all'indirizzo IP multicast e alla porta specificata (l'indirizzo IP e la porta devono essere configurati come indirizzi multicast su ogni telefono)

static int audioPlay(String where, File file)

Riproduce un file audio pre-registrato. Restituisce -1 in caso di errore, o l'indice del messaggio in coda da riprodurre.

Parametri:

where - destinazione del messaggio vocale; vedi tabella sopra

file - il file audio.

static int audioPlay(String where, String voice, String text)

Converte un messaggio di testo in un messaggio vocale utilizzando il text-to-speech installato, e lo riproduce. Restituisce -1 in caso di errore, o l'indice del messaggio in coda da riprodurre.

Parametri:

where - destinazione del messaggio vocale; vedi tabella sopra

voice - il nome della voce di text-to-speech; per maggiori dettagli consultare l'Appendice Public Announcement

text - il messaggio di testo da convertire in text-to-speech.

Data Logger

static boolean dataLoggerRefresh(String name)

Forza il data logger a fare il controllo di consistenza dei dati rispetto all'ora e la data corrente. Se necessario, esegue lo spostamento dei dati sui grafici del periodo precedente.

Parametri:

name - identificativo del data logger di riferimento

static boolean dataLoggerUpdate(String name, double value)

Fornisce un nuovo valore da processare al data logger specificato.

Parametri:

name - identificativo del data logger di riferimento

value - valore da processare

static boolean dataLoggerClear(String name)

Cancella i dati raccolti data logger specificato.

Parametri:

name - identificativo del data logger di riferimento

static boolean dataLoggerSave(String type, String[] names, String path, boolean timestamp)

Crea e/o aggiorna il file CSV specificato con i dati processati dal data logger indicato. I valori possibili per il parametro **type** sono:

- **log**: il metodo aggiungerà una riga contenente l'ultimo valore acquisito dal data logger al file specificato. Se il parametro **timestamp** è impostato a *true*, verrà aggiunto anche il timestamp corrispondente al momento di acquisizione del valore
- **stat**: aggiorna il file specificato con i valori attualmente raccolti dal data logger.

Parametri:

type - tipo di log (*log* o *stat*)

names - array di identificativi dei data logger indirizzati

path - percorso del file

timestamp - se *true* ed il parametro **type** è impostato a *log*, aggiunge una colonna di timestamp al file CSV

static boolean dataLoggerOptions(String name, String param, String value)

Imposta una determinata opzione sul data logger specificato.

Per ora l'unica opzione disponibile è "origin" usata per impostare il valore dell'asse x dei grafici legati al data logger al valore passato.

Parametri:

name - identificativo del data logger di riferimento

param - opzione da impostare ("origin" è l'unica disponibile per ora)

value - valore per l'opzione specificata

Leak Detector

L'Intelligent Leak Detector viene utilizzato per segnalare potenziali perdite di acqua o altre sostanze analizzando un generico contatore di flusso.

Il leak detector viene richiamato ogni volta che il contatore di flusso viene incrementato. Segnerà una perdita se il flusso misurato risulta relativamente costante oltre un determinato periodo di tempo.

Utilizza questo metodo a tuo rischio!

L'Intelligent Leak Detector utilizza un algoritmo di correlazione che distingue tra un flusso costante e un flusso irregolare. Un flusso relativamente costante oltre un certo periodo di tempo è ritenuto una potenziale perdita e genera un evento leak. In alcune condizioni, anche in relazione al tipo di contatore di flusso utilizzato, questo algoritmo potrebbe fallire nel riconoscere una reale perdita, o potrebbe generare delle false segnalazioni di perdita.

Utilizzare nomi differenti per implementare più leak detector indipendenti.

static void leakDetectorClear(String name)

cancella lo stato interno del leak detector. In normali applicazioni, non c'è bisogno di richiamare questa funzione, poichè il leak detector cancella automaticamente il suo stato in relazione ai dati passati.

Parametri:

name - nome del leak detector.

static void leakDetectorOptions(String name, String param, String value)

La logica del leak detector sfrutta due parametri per impostare il livello di sensibilità e l'arco temporale.

E' possibile modificare i valori di default del leak detector per adattarsi a condizioni specifiche.

Il parametro "period" cambia il periodo di tempo base. Il valore di default è di 1200 secondi.

Il leak detector genera un messaggio di avviso se la deviazione del flusso rimane costante al di sotto di una soglia di deviazione per un determinato periodo di tempo.

Il parametro "deviation" cambia la percentuale di deviazione, e il suo valore deve essere un numero compreso tra 1 e 100. Il valore di default è 50.

Valori più bassi possono provocare situazioni che attivano segnalazione di perdita solo se il flusso è molto costante nel tempo, diventando così meno suscettibile ai falsi allarmi, ma un valore troppo basso potrebbe anche perdere perdite effettive. Valori più alti possono invece generare falsi allarmi.

Il periodo di tempo e la deviazione possono essere modificati in qualunque momento, anche quando il leak detector sta processando dati.

Parametri:

name - nome del leak detector

param - impostare a "period" o "deviation"

value - valore del periodo o della deviazione, in formato stringa.

public static int leakDetectorUpdate(String name, double value)

Richiama il leak detector passando il valore di flusso aggiornato. Restituisce -1 se lo stato del leak detector non è cambiato, 0 se lo stato è cambiato a una condizione di reset, e 1 se lo stato ha modificato la condizione di perdita.

Parametri:

name - nome del leak detector

value - valore del contatore di flusso.

Mail

static int sendMail(String to, String from, String subject, String body)

Invia una email.

HSYCO SERVER invia una email o direttamente al dominio di mail server del destinatario se il parametro SmtplibName non è stato dichiarato in hsysco.ini, oppure utilizzando un account email specifico con autenticazione dell'utente e traffico criptato se il server STMP e i parametri dell'account sono stati definiti. Se il messaggio mail viene inviato direttamente al dominio di mail server del destinatario, è necessario assicurarsi che il mail server accetti email all'indirizzo di destinazione che è stato specificato e dall'indirizzo IP pubblico di HSYCO SERVER.

Questo metodo non ritenta di inviare il messaggio se il mail server di destinazione non è disponibile quando il metodo viene chiamato.

Il metodo restituisce 1 se il messaggio è stato inviato con successo, 0 se per l'indirizzo del destinatario non è disponibile un mail server valido, -1 in caso di errore generico.

Parametri:

to - indirizzo email valido del destinatario. E' anche possibile specificare il nome del server SMTP o l'indirizzo IP aggiungendo alla fine dell'indirizzo email del destinatario :<nome server name o indirizzo>.

Ad esempio: *john@example.com:192.168.1.1*

from - indirizzo email valido del mittente

subject - oggetto del messaggio

body - corpo del messaggio.

static int sendMail(String to, String from, String subject, Vector<String> body)

Invia un messaggio email, composto da testo e immagini delle telecamere, ad un singolo indirizzo, effettuando una connessione diretta al mail server del destinatario. Il messaggio può contenere più immagini registrate o in diretta.

HSYCO SERVER invia una email o direttamente al dominio di mail server del destinatario se il parametro SmtplibName non è stato dichiarato in hsyco.ini, oppure utilizzando un account email specifico con autenticazione dell'utente e traffico criptato se il server STMP e i parametri dell'account sono stati definiti. E' necessario verificare che il mail server destinatario del messaggio accetti e consideri valida la mail proveniente dall'indirizzo specificato come mittente, inviata dall'indirizzo IP pubblico di HSYCO SERVER.

Questo metodo non effettua tentativi di invio multipli e differiti nel caso l'invio della mail non vada immediatamente a buon fine.

Restituisce 1 in caso di successo, 0 se per l'indirizzo del destinatario non è disponibile un mail server valido, oppure -1 in caso di errore generico.

Parametri:

to - indirizzo email valido del destinatario. E' anche possibile specificare il nome del server SMTP o l'indirizzo IP aggiungendo alla fine dell'indirizzo email del destinatario :<nome server name o indirizzo>.

Ad esempio: *john@example.com:192.168.1.1*

from - indirizzo email valido del mittente

subject - oggetto del messaggio

body - corpo del messaggio. Gli elementi del vettore possono contenere qualunque testo, che verrà inserito nel messaggio nell'ordine del vettore, oppure dei riferimenti alle immagini, nel formato: "cam:<cameraname>[:<seconds_back>]". Ad esempio, "cam:door" invia l'ultimo frame ricevuto in diretta dalla telecamera con id "door"; "cam:door:2" invece indica il frame registrato due secondi prima

dell'ultimo frame registrato; "cam:door:0" indica l'ultimo frame registrato. Per inviare file come allegati, utilizzare il formato: "file:<file name>". Il percorso del file fa riferimento alla directory root di HSYCO.

Log

static void messageLog(String message)

Genera un messaggio informativo nel file di log.

Parametri:

message - il testo del messaggio.

static void errorLog(String message)

Genera un messaggio di errore nel file di log.

Parametri:

message - il testo del messaggio di errore.

Servizi di Rete

ping(String host, int timeout)

Verifica la raggiungibilità dell'indirizzo di rete specificato.

Restituisce *true* se raggiungibile, *false* altrimenti.

Parametri:

host - nome o indirizzo IP da verificare

timeout - timeout in millisecondi.

static int wakeOnLan(String broadcast, String address)

Invia un comando Wake on LAN sulla rete indicata dall'indirizzo di broadcast per l'interfaccia di rete identificata da un indirizzo IP o MAC.

Restituisce -1 in caso di errore, 1 in caso di successo.

Parametri:

broadcast - l'indirizzo LAN di broadcast. Ad esempio "192.168.0.255"

address - l'indirizzo IP o il MAC address della interfaccia di rete cui inviare il comando. Ad esempio un indirizzo IP "192.168.0.1" oppure un MAC address "45-ab-44-00-00-ba".

Qualora venga utilizzato l'indirizzo IP per identificare il destinatario del comando, HSYCO tenterà di determinare automaticamente il corrispondente MAC address, salvando poi in modo permanente questa associazione²⁶. Quindi, utilizzando l'indirizzo IP e solo per la prima esecuzione di questo metodo, il destinatario dovrà essere attivo con l'indirizzo IP assegnato per consentire ad HSYCO di determinare il MAC address.

²⁶ La cache di associazione tra indirizzi IP e MAC address è salvata nel file wol.data nella directory principale di HSYCO.
HSYCO 3.1.2 REL. IT 1

Metodi di Sistema

static boolean haActiveState()

Quando HSYCO è installato in configurazione ad alta affidabilità, questo metodo restituisce *true* se il sistema (master o slave) è attivo, *false* se non attivo.

Richiamato in configurazioni non ad alta affidabilità restituisce sempre *true*.

static boolean isDaylight()

Restituisce *true* se giorno, *false* se notte. Il risultato è calcolato basandosi sui dati di latitudine e longitudine impostati in `hsyco.ini` e i parametri `SunriseOffsetMinutes` e `SunsetOffsetMinutes`.

static long getNextSunrise(long now, boolean withoffset)

Questo metodo calcola l'orario dell'alba successiva all'orario impostato nel parametro *now*. L'ora dell'alba è indicata in millisecondi a partire dalla mezzanotte del 1 gennaio 1970 UTC.

Parametri:

now - il tempo di riferimento in millisecondi

withoffset - *true* per tenere in considerazione lo scostamento dall'orario dell'alba definito in *SunriseOffsetMinutes* in `hsyco.ini`, *false* per ignorare lo scostamento e ritornare all'orario di alba effettivo.²⁷

static long getNextSunset(long now, boolean withoffset)

Questo metodo calcola l'orario del tramonto successivo all'orario di riferimento impostato nel parametro *now*.

²⁷ Il calcolo dell'alba è basato sullo standard di "alba civile", corrispondente a una posizione del Sole di 6° al di sotto dell'orizzonte.

L'ora del tramonto è indicata in millisecondi a partire dalla mezzanotte del 1 gennaio 1970 UTC.

Parametri:

now - il tempo di riferimento in millisecondi

withoffset - *true* per tenere in considerazione lo scostamento dall'orario di tramonto definito in *SunsetOffsetMinutes* in *hsyco.ini*, *false* per ignorare lo scostamento e ritornare all'orario di tramonto effettivo.²⁸

static void programTimerSet(String name, int seconds)

Imposta un program timer con esecuzione dopo il numero di secondi indicato. Se è già impostato un program timer con lo stesso nome, l'azione viene ignorata.

Parametri:

name - il nome del program timer

seconds - il numero di secondi all'esecuzione del program timer.

static void programTimerClear(String name)

Cancella l'esecuzione di un program timer.

Parametri:

name - il nome del program timer.

static void programTimerReset(String name, int seconds)

Imposta un program timer con esecuzione dopo il numero di secondi indicato. Se è già impostato un program timer con lo stesso nome, il numero di secondi di attesa viene reimpostato al nuovo valore.

²⁸ Il calcolo del tramonto è basato sullo standard di "tramonto civile", corrispondente a una posizione del Sole di 6° al di sotto dell'orizzonte.

Parametri:

name - il nome del program timer

seconds - il numero di secondi all'esecuzione del program timer.

static void programTimerRepeat(String name, int seconds)

Imposta un program timer con esecuzione ripetuta con intervallo pari al numero di secondi indicato. Se è già impostato un program timer con lo stesso nome, la durata dell'intervallo viene reimpostata al nuovo valore.

Parametri:

name - il nome del program timer

seconds - la durata in secondi dell'intervallo tra esecuzioni successive del program timer.

static void sleep(long millis)

Causa una attesa per il numero di millisecondi specificato.

Parametri:

millis - tempo di attesa in millesimi di secondo.

static void schedulerRegister(String groupname, String schedulename, int interval)

Definisce ed inizializza un nuovo scheduler. Dal momento dell'inizializzazione, verrà periodicamente eseguita la chiamata al metodo SchedulerEvent(), ad intervalli definiti dal parametro interval.

Parametri:

groupname - nome del gruppo

schedulename - nome dello scheduler

interval - intervallo di tempo, in millisecondi

static void SchedulerRemove(String groupname, String schedulename)

Cancella uno scheduler.

Parametri:

groupname - nome del gruppo

schedulename - nome dello scheduler

static void user(String name, String param)

Questo metodo causa l'esecuzione dell'evento USER e del metodo userCommand(String name, String param). Può essere utilizzato per eseguire funzioni tra Java ed EVENTS.

Parametri:

name - il parametro name passato a userCommand()

param - il parametro param passato a userCommand()

static String varGet(String name)

Restituisce il valore di una variabile oppure null se non ci sono variabili definite con il nome specificato. Le variabili di programma possono essere impostate con il metodo varSet() in user.java o nell'ambiente EVENTS. Per leggere una variabile impostata in EVENTS, è necessario che il nome utilizzato in varGet() inizi con il carattere \$.

Le variabili il cui nome termina con il carattere ! sono persistenti ed il valore impostato viene mantenuto anche quando HSYCO viene riavviato.

Parametri:

name - il nome della variabile. I caratteri maiuscoli e minuscoli sono considerati uguali.

static void varSet(String name, String value)

Imposta una variabile al valore indicato. Le variabili di programma impostate con `varSet()` possono essere lette con il metodo `varGet()` e sono inoltre accessibili in `EVENTS`, ma solo se il nome impostato in `varSet()` inizia con il carattere `$`. Se il nome di una variabile non inizia con `$`, questa potrà essere letta solo in `user.java` con il metodo `varGet()`, ma non in `EVENTS`. Le variabili il cui nome termina con il carattere `!` sono persistenti ed il valore impostato viene mantenuto anche quando `HSYCO` viene riavviato.

Parametri:

name - il nome della variabile. I caratteri maiuscoli e minuscoli sono considerati uguali
value - il valore della variabile.

static String uiGet(String id, String attr)

Restituisce il valore corrente di un attributo per un oggetto, impostato con il metodo `uiSet()` o con l'azione `UISET` in `events.txt`. Restituisce *null* se l'attributo dell'oggetto non è stato impostato precedentemente.

Parametri:

id - id dell'oggetto
attr - nome dell'attributo
value - valore dell'attributo.

static void uiSet(String id, String attr, String value)

Consente di modificare dinamicamente alcuni attributi degli oggetti identificabili, come (text), (user), (link) ed altri. id è il nome unico che identifica l'oggetto, specificato nel file `index.hsm` con l'estensione `lid` dopo il tipo dell'oggetto, ad esempio:

```
(text!thisobject ... )
```


Parametri:

id - id dell'oggetto

attr - nome dell'attributo; vedi tabella seguente

value - valore dell'attributo; vedi tabella seguente.

tipo oggetto	attributo	valore
project	page	imposta l'id al nome della directory del file index.hsm che si vuole controllare, o a "*" per controllare tutti i client Web. Impostando value all'id della pagina o a "menu" viene forzata la visualizzazione di quella pagina su tutti i browser che hanno il menu visibile nel momento in cui il comando viene eseguito. Se value è una stringa vuota il comando eseguito riporta alla pagina precedente.
	lock	imposta l'id al nome della directory del file index.hsm che si vuole controllare, o a "*" per controllare tutti i client Web. Impostando value all'id della pagina o a "menu" viene forzata la visualizzazione di quella pagina su tutti i browser che hanno il menù visibile nel momento in cui il comando viene eseguito, ma anche per i menù visualizzati successivamente all'esecuzione del comando. Inoltre viene bloccata la navigazione dell'utente. Se invece value è una stringa vuota la navigazione viene sbloccata.
	pageback	imposta l'id al nome della directory del file index.hsm che si vuole controllare, o a "*" per controllare tutti i client connessi. Impostare value all'id della pagina per forzare i client che stanno visualizzando la pagina corrente a quella precedente. Impostare value all'id del pop-up per forzare i client che hanno quel pop-up aperto a chiuderlo
page	blink	<i>false</i> : tutti gli oggetti della pagina smettono di lampeggiare
background	img	nome del file personalizzato con l'immagine dell'oggetto salvato nella directory www/img, o l'URL completo (compreso di https:// iniziale) per mostrare un'immagine da un sito remoto
camera	camera	id della telecamera da visualizzare
camerapanel	visible	<i>true</i> : visibile; <i>false</i> : non visibile

tipo oggetto	attributo	valore
camerapanel	blink	<i>true</i> : lampeggio lento; <i>slow</i> : lampeggio lento; <i>fast</i> : lampeggio veloce; <i>false</i> : stop lampeggio
	pos	la posizione dell'oggetto
chart	visible	<i>true</i> : visibile; <i>false</i> : non visibile
	blink	<i>true</i> : lampeggio lento; <i>slow</i> : lampeggio lento; <i>fast</i> : lampeggio veloce; <i>false</i> : stop lampeggio
	pos	la posizione dell'oggetto
	type	tipo di grafico; valori ammessi: <i>bars</i> , <i>points</i> , <i>gauge</i> . Il chart di tipo gauge supporta solo i seguenti attributi: <i>value</i> , <i>orientation</i> , <i>valuerange</i> , <i>bgcolor</i> , <i>barcolor</i> , <i>barborder</i> ; l'attributo <i>valuerange</i> è necessario per una corretta rappresentazione grafica del valore
	orientation	orientamento verticale o orizzontale dell'asse dei valori; valori ammessi: <i>vertical</i> , <i>horizontal</i>
	value	il valore, solo per chart di tipo gauge
	values	la serie di valori separati da virgola, per chart di tipo bars e points. E' possibile omettere alcuni valori della serie (es. <i>values="1, 3, 2, ,4, , , 5"</i>)
	valuerange	valore minimo e massimo separati da virgola. Indicano il range del grafico. Es. <i>"-100,100"</i> (con <i>orientation vertical</i>) visualizzerà un grafico con valori da -100 a 100 sull'asse y
	axislabels	etichette sull'asse x (se <i>orientation</i> è <i>vertical</i>) o sull'asse y (se <i>orientation</i> è <i>horizontal</i>). Non deve necessariamente corrispondere al numero di valori in <i>values</i> e può contenere label vuote (in tal caso vengono visualizzate solo le tacche)
	drawaxis	specifica se gli assi (comprese le tacche e le label) vengono disegnati o meno; valori possibili: <i>true</i> o <i>false</i>
	origin	sposta l'origine ad un valore diverso da 0 (ad esempio <i>origin=3.2</i>)
	axisoffset	l'offset dell'asse (y se <i>vertical</i> , x se <i>horizontal</i>) in percentuale o come indice della serie (es. <i>"50%"</i> o <i>"3"</i>)
	baroffset	sposta le barre o i punti del numero di pixel specificato, rendendo più semplice la sovrapposizione di più chart
	notches	numero di tacche sull'asse x (se <i>orientation</i> è <i>horizontal</i>) o sull'asse y (se <i>orientation</i> è <i>vertical</i>)

tipo oggetto	attributo	valore
	spacing	spaziatura in percentuale fra le barre (valido solo per la modalità <i>bars</i>)
	pointsize	dimensione dei "punti" (valido solo per la modalità <i>points</i>)
	vlabelsstyle	stile delle value label; valori possibili: <i>inside</i> , <i>outside</i> , <i>none</i>
	bgcolor	colore dello sfondo in codice HTML
	barcolor	serie di colori HTML, uno per ogni valore in values
	barborder	visibilità del bordo delle barre/punti; valori possibili: <i>true</i> o <i>false</i>
	axiscolor	colore HTML degli assi
	notchcolor	colore HTML delle tacche sugli assi
	labelcolor	colore HTML delle etichette sugli assi e sulle barre
container	visible	<i>true</i> : visibile; <i>false</i> : non visibile
	blink	<i>true</i> : lampeggio lento; <i>slow</i> : lampeggio lento; <i>fast</i> : lampeggio veloce; <i>false</i> : stop lampeggio
	pos	la posizione dell'oggetto
dlink	visible	<i>true</i> : visibile; <i>false</i> : non visibile
	blink	<i>true</i> : lampeggio lento; <i>slow</i> : lampeggio lento; <i>fast</i> : lampeggio veloce; <i>false</i> : stop lampeggio
	pos	la posizione dell'oggetto
	text	il testo visualizzato
	color	<i>r</i> : rosso, <i>g</i> : verde, <i>b</i> : blu, <i>gr</i> : grigio, <i>y</i> : giallo
image imagelink	visible	<i>true</i> : visibile; <i>false</i> : non visibile
	blink	<i>true</i> : lampeggio lento; <i>slow</i> : lampeggio lento; <i>fast</i> : lampeggio veloce; <i>false</i> : stop lampeggio
	pos	la posizione dell'oggetto
	img	nome del file contenuto nella directory <i>www/img</i> , o l'URL completo (compreso di <i>https://</i> iniziale) per mostrare un'immagine da un sito remoto
	text	il testo visualizzato
input keypad	visible	<i>true</i> : visibile; <i>false</i> : non visibile
	blink	<i>true</i> : lampeggio lento; <i>slow</i> : lampeggio lento; <i>fast</i> : lampeggio veloce; <i>false</i> : stop lampeggio
	pos	la posizione dell'oggetto
	text	il testo visualizzato
	color	il colore del testo; utilizzare i codici di colore standard HTML
	size	la grandezza dei caratteri in pixel

tipo oggetto	attributo	valore
	width	larghezza, in pixel
	height	altezza, in pixel
	weight	lo spessore del testo; valori possibili: <i>lighter</i> , <i>normal</i> , <i>bold</i> , <i>bolder</i>
	style	lo stile del testo; valori possibili: <i>normal</i> , <i>italic</i> , <i>oblique</i>
link	visible	<i>true</i> : visibile; <i>false</i> : non visibile
linkmini	blink	<i>true</i> : lampeggio lento; <i>slow</i> : lampeggio lento; <i>fast</i> : lampeggio veloce; <i>false</i> : stop lampeggio
linkmicro	pos	la posizione dell'oggetto
	text	il testo visualizzato
	color	<i>r</i> : rosso, <i>g</i> : verde, <i>b</i> : blu, <i>gr</i> : grigio, <i>y</i> : giallo
marquee	visible	<i>true</i> : visibile; <i>false</i> : non visibile
	blink	<i>true</i> : lampeggio lento; <i>slow</i> : lampeggio lento; <i>fast</i> : lampeggio veloce; <i>false</i> : stop lampeggio
	pos	la posizione dell'oggetto
	text	il testo visualizzato
	color	il colore del testo; utilizzare i codici di colore standard HTML
	size	la grandezza dei caratteri, in pixel
	width	la larghezza del campo di testo, in pixel
	align	l'allineamento del testo; valori possibili: <i>left</i> , <i>right</i> , <i>center</i> , <i>justified</i>
	weight	lo spessore del testo; valori possibili: <i>lighter</i> , <i>normal</i> , <i>bold</i> , <i>bolder</i>
	style	lo stile del testo; valori possibili: <i>normal</i> , <i>italic</i> , <i>oblique</i>
	height	altezza dell'area di scorrimento
	paused	<i>true</i> : ferma lo scorrimento; <i>false</i> : riparte
panel	visible	<i>true</i> : visibile; <i>false</i> : non visibile
	blink	<i>true</i> : lampeggio lento; <i>slow</i> : lampeggio lento; <i>fast</i> : lampeggio veloce; <i>false</i> : stop lampeggio
	pos	la posizione dell'oggetto
	color	<i>r</i> : rosso, <i>g</i> : verde, <i>b</i> : blu, <i>gr</i> : grigio, <i>y</i> : giallo. Impostare il colore a una stringa vuota equivale a impostare il colore di default del pannello (trasparente)
slider	visible	<i>true</i> : visibile; <i>false</i> : non visibile

tipo oggetto	attributo	valore
	blink	<i>true</i> : lampeggio lento; <i>slow</i> : lampeggio lento; <i>fast</i> : lampeggio veloce; <i>false</i> : stop lampeggio
	value	<i>on</i> , <i>off</i> , un valore da 0% a 100%, da 0.00 a 1.00, da 0/<max> a <max>/<max>
submit	visible	<i>true</i> : visibile; <i>false</i> : non visibile
submitmini	blink	<i>true</i> : lampeggio lento; <i>slow</i> : lampeggio lento; <i>fast</i> : lampeggio veloce; <i>false</i> : stop lampeggio
submitmicro	pos	la posizione dell'oggetto
	text	il testo visualizzato
	color	<i>r</i> : rosso, <i>g</i> : verde, <i>b</i> : blu, <i>gr</i> : grigio, <i>y</i> : giallo
submitimage	visible	<i>true</i> : visibile; <i>false</i> : non visibile
	blink	<i>true</i> : lampeggio lento; <i>slow</i> : lampeggio lento; <i>fast</i> : lampeggio veloce; <i>false</i> : stop lampeggio
	pos	la posizione dell'oggetto
	img	nome del file immagine contenuto nella directory <code>www/img</code>
text	visible	<i>true</i> : visibile; <i>false</i> : non visibile
	blink	<i>true</i> : lampeggio lento; <i>slow</i> : lampeggio lento; <i>fast</i> : lampeggio veloce; <i>false</i> : stop lampeggio
	pos	la posizione dell'oggetto
	text	il testo visualizzato
	color	il colore del testo; utilizzare i codici di colore standard HTML
	size	la grandezza dei caratteri, in pixel
	width	la larghezza del campo di testo, in pixel
	align	l'allineamento del testo; valori possibili: <i>left</i> , <i>right</i> , <i>center</i> , <i>justified</i>
	weight	lo spessore del testo; valori possibili: <i>lighter</i> , <i>normal</i> , <i>bold</i> , <i>bolder</i>
	style	lo stile del testo; valori possibili: <i>normal</i> , <i>italic</i> , <i>oblique</i>
user	visible	<i>true</i> : visibile; <i>false</i> : non visibile
usermini	blink	<i>true</i> : lampeggio lento; <i>slow</i> : lampeggio lento; <i>fast</i> : lampeggio veloce; <i>false</i> : stop lampeggio
usermicro	pos	la posizione dell'oggetto
	text	il testo visualizzato
	color	<i>r</i> : rosso, <i>g</i> : verde, <i>b</i> : blu, <i>gr</i> : grigio, <i>y</i> : giallo
userimage	visible	<i>true</i> : visibile; <i>false</i> : non visibile

tipo oggetto	attributo	valore
	blink	<i>true</i> : lampeggio lento; <i>slow</i> : lampeggio lento; <i>fast</i> : lampeggio veloce; <i>false</i> : stop lampeggio
	pos	la posizione dell'oggetto
	img	nome del file immagine contenuto nella directory <code>www/img</code>
	text	il testo visualizzato
userrgb	visible	<i>true</i> : visibile; <i>false</i> : non visibile
	blink	<i>true</i> : lampeggio lento; <i>slow</i> : lampeggio lento; <i>fast</i> : lampeggio veloce; <i>false</i> : stop lampeggio
	pos	la posizione dell'oggetto
	text	il testo visualizzato
	color	il colore del tasto; utilizzare i codici di colore standard HTML
video	visible	<i>true</i> : visibile; <i>false</i> : non visibile
	blink	<i>true</i> : lampeggio lento; <i>slow</i> : lampeggio lento; <i>fast</i> : lampeggio veloce; <i>false</i> : stop lampeggio
	pos	la posizione dell'oggetto
	play/pause	riproduzione/pausa del video
	mode	modalità di riproduzione: <i>stop</i> , <i>auto</i> e <i>loop</i>

static void powerSet(int power)

Imposta la variabile di stato relativa alla potenza elettrica. Il valore di potenza viene mostrato nella barra di stato o di navigazione dell'interfaccia Web. Oltre a impostare il valore di potenza, questo metodo richiama anche l'esecuzione del metodo di callback `powerEvent()` e l'evento `POWER` in `EVENTS`.

Parametri:

power - la potenza in Watt.

Appendici

Data Logger

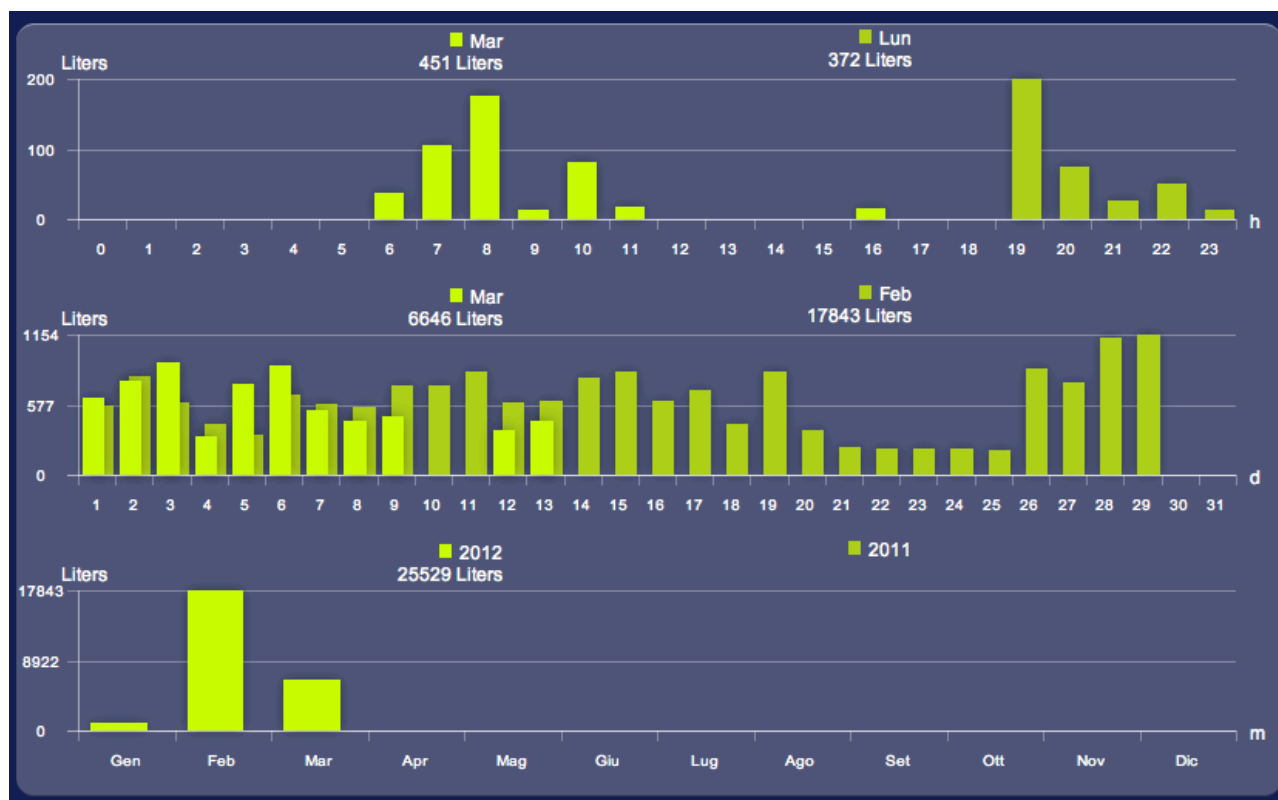
I data logger permettono di raccogliere, processare e visualizzare dati statistici sulla variazione di una sequenza di dati. Possono essere utilizzati per creare grafici periodici dell'andamento di tale valore e/o per creare file di log in formato CSV.

Un data logger raccoglie dati e li raggruppa al fine di calcolare l'andamento per ogni ora (o gruppi di ore) del giorno corrente e passato, per ogni giorno del mese corrente e passato e per ogni mese dell'anno corrente e passato. Inoltre, consente di salvare i dati processati in un file di log.

Vi sono due tipi di data logger:

- **counter**: adatto all'elaborazione di un valore incrementale (es. consumo o produzione energetica). Calcola la variazione (delta) del valore per ogni intervallo di tempo rispetto al precedente. Inoltre, permette di specificare fasce orarie e le relative tariffe per calcolare, ad esempio, i costi relativi ai consumi energetici.
- **range**: mirato al calcolo del massimo, del minimo e della media di un valore oscillante (es. una temperatura o un valore di umidità misurati) per ogni intervallo di tempo.

Quando un data logger viene definito in *hsyco.ini*, è possibile passare i dati acquisiti utilizzando una semplice azione in EVENTS, oppure un metodo Java, e visualizzare i dati statistici con l'oggetto (datalogger).



HSYCO può anche opzionalmente²⁹ gestire automaticamente l'aggiornamento di chart distinte, così da consentire una più ampia possibilità di personalizzazione della presentazione grafica.

Data Logger di tipo Counter

Il primo passo per creare un data logger di tipo counter è inizializzarlo in *hsyco.ini*. Per una descrizione completa di tutti i parametri e le opzioni disponibili, far riferimento alla sezione sui data logger nel capitolo di configurazione. Qui dichiariamo un data logger di tipo counter denominato “energy” lasciando i parametri di default.

In *hsyco.ini* aggiungere le righe seguenti:

```
DataLoggers = energy
DataLoggers.energy.Type = counter
```

²⁹ L'aggiornamento dei chart è automatico se `DataLoggers.<id>.UseCharts = true`.

Adesso dobbiamo passare al data logger il valore incrementale da processare. Assumendo di avere un I/O server denominato “elektro” che fornisce il valore corrispondente al consumo energetico, scriviamo le seguenti righe in *events.txt*:

```
IO elektro : DATALOGGER energy = IO elektro,
            DATALOGGER energy = FILE LOG energy.csv
```

Ogni volta che l'I/O server fornisce un nuovo valore, il data logger verrà aggiornato ed una riga di log verrà aggiunta al file “energy.csv”.

Infine, per visualizzare graficamente i dati acquisiti dal data logger, è sufficiente utilizzare l'oggetto (datalogger), ad esempio:

```
(datalogger energy; x80y60; 820; 500; kWh;)
```

Per una lista completa di tutte le azioni disponibili, far riferimento alla sezione relativa ai data logger nel capitolo di programmazione ad eventi. Le stesse operazioni possono essere eseguite usando codice Java (sezione Data Logger nel capitolo di programmazione Java).

Aggiungendo in *hsyco.ini* il parametro:

```
DataLoggers.energy.UseCharts = true
```

HSYCO creerà opzionalmente un set di grafici (charts) con i seguenti ID:

```
energy.counter.chart.day (grafico del giorno corrente)
energy.counter.chart.day.past (grafico del giorno precedente)
energy.counter.chart.month (grafico del mese corrente)
energy.counter.chart.month.past (grafico del mese passato)
energy.counter.chart.year (grafico dell'anno corrente)
energy.counter.chart.year.past (grafico dell'anno passato)
```

Quindi, aggiungendo semplicemente un oggetto chart nella nostra interfaccia Web e impostando il suo ID ad uno dei valori elencati sopra, si avrà un grafico automaticamente aggiornato che mostra i dati relativi al periodo specificato.

Inoltre, impostando il campo ID di un oggetto testuale (es. text) ad uno dei suddetti ID estesi dal suffisso “.total” (es. *energy.counter.chart.day.total*) si avrà un campo che visualizza la variazione totale del valore durante il periodo di riferimento.

Un altro esempio potrebbe essere la creazione di un data logger per monitorare le spese relative ai consumi energetici. A tal fine, dobbiamo inizializzare un nuovo data logger di tipo counter e dichiarare le tariffe delle fasce orarie nel file *dataloggers.ini*.

In *hsyco.ini* definiamo il nuovo data logger:

```
DataLoggers = cost
DataLoggers.cost.Type = counter
```

Poi, facciamo in modo che l'I/O server "elektro" aggiorni anche il nuovo data logger. Quindi, in *events.txt*, aggiungiamo le seguenti righe:

```
IO elektro OR TIME : DATALOGGER cost = IO elektro
```

Da notare come questa volta aggiorniamo il data logger non solo quando un nuovo valore è disponibile, ma anche ogni minuto. Ciò è necessario per avere un calcolo accurato delle spese.

Ora definiamo le nostre fasce orarie nel file *dataloggers.ini*, ad esempio:

```
cost; 0; 00:00-11:59; *; *; 0.3
cost; 1; 12:00-16:59; *; *; 1.4
cost; 2; 17:00-23:59; *; *; 0.8
```

Queste regole impongono che ogni giorno da mezzanotte alle 11:59:59 siamo nella fascia 0 e la tariffa da applicare è 0.3, dalle 12:00:00 fino alle 16:59:59 siamo in fascia 1 ed il rate corrispondente è 1.4, successivamente saremo in fascia 2 con una tariffa pari a 0.8.

Possiamo aggiungere qualche eccezione aggiungendo delle righe prima di tali regole:

```
cost; 0; *; *, 2000/12/10-2000/12/31; 0.3
cost; 2; *; 67; *, 0.8
cost; 0; 00:00-11:59; *; *; 0.3
cost; 1; 12:00-16:59; *; *; 1.4
cost; 2; 17:00-23:59; *; *; 0.8
```

Le regole precedenti saranno ancora applicate, con l'eccezione dei Sabati (6) e delle Domeniche (7) quando, durante tutto il giorno, la fascia 2 sarà applicata.

Inoltre, durante il periodo tra il 10 ed il 31 Dicembre del 2000, verrà applicata la tariffa della fascia 0 indipendentemente dal giorno della settimana o l'ora del giorno.

Per una descrizione completa di come scrivere regole per le fasce orarie, far riferimento alla sezione sul file *dataloggers.ini*.

A questo punto, utilizzando l'oggetto (datalogger) descritto prima otterremo automaticamente la visualizzazione di tutti i dati acquisiti, con la possibilità di filtrare la visualizzazione in base alle tariffe impostate.

In alternativa possiamo aggiungere alla nostra interfaccia Web gli oggetti chart che mostrano le spese periodiche, semplicemente impostando l'ID appropriato tra i seguenti:

```
cost.counter.chart.day
cost.counter.chart.day.past
cost.counter.chart.month
cost.counter.chart.month.past
cost.counter.chart.year
cost.counter.chart.year.past
```

Se desiderato, possiamo avere grafici mensili e annuali separati per ogni fascia definita. Possiamo far ciò impostando il parametro *SeparateCharts* a *true* in *hsyco.ini*:

```
DataLoggers.cost.UseCharts = true
DataLoggers.cost.SeparateCharts = true
```

Così facendo, HSYCO metterà a disposizione i seguenti oggetti chart (considerando che abbiamo definito le fasce 0, 1 e 2 in *dataloggers.ini*):

```
cost.counter.chart.day
cost.counter.chart.day.past
cost.counter.slot0.chart.month
cost.counter.slot0.chart.month.past
cost.counter.slot1.chart.month
cost.counter.slot1.chart.month.past
cost.counter.slot2.chart.month
cost.counter.slot2.chart.month.past
cost.counter.slot0.chart.year
```



```
cost.counter.slot0.chart.year.past
cost.counter.slot1.chart.year
cost.counter.slot1.chart.year.past
cost.counter.slot2.chart.year
cost.counter.slot2.chart.year.past
```

Inoltre, possiamo avere un report dettagliato di tutti i dati processati specificando il percorso di un file di log in *hsyco.ini*:

```
DataLoggers = cost
DataLoggers.cost.Type = counter
DataLoggers.cost.RatesLogFile = counterRates/log.csv
```

In questo modo verrà creato un file CSV nel quale vi sarà una riga per ogni aggiornamento del data logger, divisa nelle seguenti colonne:

nome_data_logger, ora_aggiornamento, valore_passato, costo_calcolato, id_fascia, tariffa

Data Logger di tipo Range

Per dichiarare un data logger di tipo range, denominato ad esempio “temperature”, aggiungiamo queste righe in *hsyco.ini*:

```
DataLoggers = temperature
DataLoggers.temperature.Type = range
```

N.B. La media è calcolata come la somma dei valori divisa per il numero di record. E’ quindi consigliato di fornire dati periodicamente per avere un calcolo corretto.

Adesso, facciamo in modo che il data logger elabori le temperature misurate da una sonda identificata dal server I/O “temp”.

Specifichiamo dunque in linguaggio EVENTS che ogni minuto il data logger venga aggiornato, inoltre registriamo il valore misurato nel file di log “temp.csv”:

```
TIME : DATALOGGER temperature = IO temp,
      DATALOGGER temperature = FILE LOG temp.csv
```


La scala dei grafici creati sarà automaticamente adattata a spaziare dal valore minimo misurato a quello massimo ed i grafici appartenenti allo stesso periodo verranno allineati. Se, per esempio, volessimo solamente mostrare i grafici che visualizzano i valori minimi, potrebbe essere meglio non allineare tale grafico agli altri. Per far ciò possiamo imporre in *hsyco.ini* di modificare la scala dei grafici dello stesso periodo indipendentemente:

```
DataLoggers.temperature.PeriodAlign = false
```

Il valore di origine dei grafici è impostato di default in corrispondenza del valore minimo. Si potrebbe preferire avere l'origine in corrispondenza di un valore specifico (ad esempio zero) così da evidenziare la differenza tra i valori al di sopra ed al di sotto di tale soglia. Possiamo far ciò impostando l'opzione *Origin*:

```
DataLoggers.temperature.Origin = 0
```

Infine, per visualizzare graficamente i dati acquisiti dal data logger, è sufficiente utilizzare l'oggetto (datalogger), ad esempio:

```
(datalogger temperature; x80y60; 820; 500; C;)
```

Aggiungendo in *hsyco.ini* il parametro:

```
DataLoggers.temperature.UseCharts = true
```

HSYCO metterà a disposizione oggetti chart con i seguenti ID:

temperature.min.chart.day (grafico dei valori minimi del giorno corrente)
temperature.min.chart.day.past (grafico dei valori minimi del giorno passato)
temperature.avg.chart.day (grafico dei valori medi del giorno corrente)
temperature.avg.chart.day.past (grafico dei valori medi del giorno passato)
temperature.max.chart.day (grafico dei valori massimi del giorno corrente)
temperature.max.chart.day.past (grafico dei valori massimi del giorno passato)

temperature.min.chart.month (grafico dei valori minimi del mese corrente)
temperature.min.chart.month.past (grafico dei valori minimi del mese passato)
temperature.avg.chart.month (grafico dei valori medi del mese corrente)
temperature.avg.chart.month.past (grafico dei valori medi del mese passato)
temperature.max.chart.month (grafico dei valori massimi del mese corrente)
temperature.max.chart.month.past (grafico dei valori massimi del mese passato)

temperature.min.chart.year (grafico dei valori minimi dell'anno corrente)

temperature.min.chart.year.past (grafico dei valori minimi dell'anno passato)

temperature.avg.chart.year (grafico dei valori medi dell'anno corrente)

temperature.avg.chart.year.past (grafico dei valori medi dell'anno passato)

temperature.max.chart.year (grafico dei valori massimi dell'anno corrente)

temperature.max.chart.year.past (grafico dei valori massimi dell'anno passato)

Così come per i data logger counter, apponendo il suffisso “.total” ai precedenti ID e assegnando tale ID ad un oggetto testuale, risulterà in un campo che visualizza il minimo, massimo o media complessivi del periodo di riferimento.

Configurazione delle Telecamere

HSYCO interagisce con le telecamere in tre modi:

1. utilizzando un url definito nel file hsyco.ini acquisisce i frame di ciascuna telecamera, solo quando necessario³⁰ e con la frequenza richiesta;
2. riceve dalla telecamera, sotto forma di richiesta HTTP inviata dalla telecamera al server Web di HSYCO, le segnalazione degli eventi di motion detection;
3. invia comandi di movimento e messa a fuoco alle telecamere con supporto PTZ.

Configurazione Utenti

E' possibile effettuare la configurazione delle telecamere in modalità anonima o accedendo a ciascuna telecamera con user e password. All'interno del file hsyco.ini è possibile specificare user e password distinti per ciascuna telecamera.

Acquisizione dei Frame

E' necessario che la telecamera sia abilitata a servire richieste di frame JPEG provenienti dal server HSYCO. E' possibile specificare, nel file hsyco.ini, user e password distinti per ciascuna telecamera.

³⁰ Per ridurre il carico di elaborazione sulle telecamere ed il traffico di rete, HSYCO acquisisce frame da ciascuna telecamera solo nei seguenti casi: 1) in seguito a eventi di motion detection; 2) quando da interfaccia Web viene richiesta la visualizzazione di una telecamera; 3) su richieste di registrazione dal codice Java personalizzato (user.class).

Motion Detection

Per ricevere segnalazioni di eventi di motion detection, HSYCO prevede che vengano generate da parte delle telecamere richieste HTTP verso il Web Server di HSYCO, nel seguente formato:

http://192.168.0.50/x/camerarec?camera=<name>

oppure:

http://192.168.0.50/x/camerarec?camera=<name>&zone=<id>

Il separatore tra i parametri *camera* e *zone* può essere “&”, “:” oppure “;”.

<name> è il nome della telecamera, corrispondente ad una delle telecamere elencate nel parametro *Cameras* in *hsyco.ini*.

<id> è un parametro opzionale che indica la zona dell'area di motion attiva. Non è necessario che sia specificato per le funzioni base di registrazione, ma viene passato come parametro al metodo Java in *user.class* associato agli eventi di motion detection e può quindi essere utilizzato a livello applicativo.

Ad esempio:

```
http://192.168.0.50/x/camerarec?camera=back
```

```
http://192.168.0.50/x/camerarec?camera=back&zone=1
```

Controllo PTZ

E' necessario che la telecamera sia abilitata al controllo PTZ.

Configurazione di Server con Porte Seriali

HSYCO supporta il controllo di porte seriali remote attraverso alcuni dispositivi di rete, come gli IRTrans, i Controller I/O e le ERxx di HW Group, o generici gateway seriali che supportano la comunicazione bidirezionale di una porta seriale attraverso una connessione TCP.

IRTrans

Gli IRTrans supportano solo il traffico seriale in uscita. Per configurare una porta comm per l'invio di dati ad una porta seriale di un dispositivo IRTrans, è necessario impostare il tipo di comm port a "ir", l'id con l'id dell'IRTrans, e impostare i parametri di handshake della porta seriale.

Ad esempio:

```
IRTrans = theater
IRTransIP.theater = 192.168.0.130
CommPorts = serial
CommPort.serial.Id = theater
CommPort.serial.Type = ir
CommPort.serial.Params = 115200,8,1,0,0
```

Controller HW Group

I Controller I/O e le ERxx di HW Group supportano il traffico seriale bidirezionale. Per configurare una porta comm per l'invio di dati ad una porta seriale di un dispositivo HW Group, è necessario impostare il tipo di comm port a "io", l'id con l'id del server I/O del controller HW Group, e impostare i parametri di handshake della porta seriale.

Ad esempio:

```
ioServers = er84
ioServersIP.er84 = 192.168.1.199
ioServersType.er84 = HWGIO
CommPorts = serial
```



```
CommPort.serial.Id = er84  
CommPort.serial.Type = io
```

Generico Gateway Seriale

HSYCO supporta generici gateway seriali che forniscono una comunicazione seriale bidirezionale attraverso una connessione TCP. Per configurare una comm port per l'invio di dati a una porta seriale di un gateway seriale, è necessario impostare il tipo della comm port a "server" e impostare l'indirizzo IP del gateway e il numero di porta TCP.

I parametri di handshake della porta seriale devono essere impostati sul gateway stesso utilizzando il proprio software applicativo.

Ad esempio:

```
CommPorts = eki  
CommPort.eki.Type = server  
CommPort.eki.IP = 192.168.0.187  
CommPort.eki.Port = 5300
```

HSYCO supporta la modalità di fail-over in alta affidabilità per gateway seriali.

E' necessario abilitare il fail-over aggiungendo un secondo indirizzo IP al parametro IP:

```
CommPorts = eki  
CommPort.eki.Type = server  
CommPort.eki.IP = 192.168.0.187, 192.168.0.188  
CommPort.eki.Port = 5300
```

HSYCO proverà a stabilire una connessione utilizzando inizialmente il primo indirizzo IP dichiarato. Qualora si verificasse un errore di comunicazione, passerà automaticamente all'indirizzo dichiarato in fail-over.

In questa configurazione, il metodo Java `closeComm(String portName)` può essere utilizzato per forzare una riconnessione alla nuova chiamata del metodo `readComm()` o del metodo `writeComm()`, in modo tale che ci sia uno scambio tra l'IP corrente e quello in fail-over, nel caso in cui l'IP corrente non risponda.

Servizio di Localizzazione Wi-Fi

Per il corretto funzionamento dei servizi di localizzazione dei client connessi in LAN Wi-Fi è necessario definire i parametri relativi nel file `hsyco.ini`, cioè ***LocationBases*** con l'elenco di tutti gli Access Point in grado di fornire informazioni di localizzazione e ***LocationBaseIP.<nomeAP>*** con l'indirizzo IP di ciascun Access Point.

Deve inoltre essere abilitato il server SYSLOG, specificando la porta IP di ascolto³¹, con il parametro ***SysLogServerPort=<port>***.

Infine è necessario configurare ciascun Access Point in modo che sia abilitato il trasferimento dei messaggi di log generati verso il SYSLOG server all'indirizzo IP del server HSYCO.

³¹ La porta standard per il servizio SYSLOG su protocollo UDP è la 514.

Servizi di Telefonia

HSYCO può ricevere le segnalazioni di chiamata dai sistemi di telefonia. Nel caso di PBX *Grandstream* deve essere abilitato il server SYSLOG, come per i servizi di localizzazione, ed il PBX deve essere configurato per inviare i log a livello **Info** al SYSLOG server all'indirizzo IP del server HSYCO.

Per gli altri sistemi, HSYCO prevede che vengano generate da parte di ciascun dispositivo richieste HTTP verso il Web Server di HSYCO, nel seguente formato:

http://192.168.0.50/x/vcall/<from>/<to>

oppure:

http://192.168.0.50/x/vcall?from=<from>&to=<to>

Ad esempio, la richiesta HTTP:

`http://192.168.0.50/x/vcall/21/25`

viene interpretata come una notifica di chiamata dal numero interno 21 al numero 25 e causa l'esecuzione del metodo callback *PBXCallEvent()* e dell'evento *PBX* in EVENTS.

Public Announcement

HSYCO può riprodurre annunci vocali come file pre-registrati o utilizzando un motore text-to-speech installato che converte il testo in audio.

Gli annunci vocali possono essere inviati al browser Web, oppure tramite l'uscita audio line-out del server, attraverso l'altoparlante interno o l'uscita audio delle telecamere Axis e l'altoparlante dei telefoni SNOM o dei dispositivi PA.

Riproduzione File Audio

Quando viene riprodotto un file audio pre-registrato, il formato deve essere supportato dalla destinazione audio che si intende utilizzare.

Destinazione	Formati Audio
speaker	la maggior parte dei formati audio supportati da Linux o Mac OS X funzionano correttamente
web	utilizzare un formato audio supportato dal Web browser (solamente le ultime versioni di Chrome e Firefox sono supportate)
axis	utilizzare il formato uLaw, 8 KHz, 8 bit, mono, file WAV
snom	utilizzare il formato uLaw, 8 KHz, 8 bit, mono, file WAV

Il Motore Text-to-Speech

HSYCO supporta diversi motori text-to-speech su Linux e Mac OS X.

Su Linux, il motore di default è eSpeak³², con la possibilità di utilizzare il motore opzionale Acapela che offre una vasta gamma di voci in altissima qualità.

Su Mac OS X, HSYCO utilizza il servizio di tex-to-speech integrato e tutto il pacchetto voci disponibile.

³² Per maggiori dettagli sul software eSpeak visita il sito <http://espeak.sourceforge.net/>

eSpeak

Se si utilizza eSpeak, impostare il nome della voce tra quelle supportate. Nella tabella seguente vengono mostrate solo alcune voci, per la lista completa consultare la documentazione on-line di eSpeak.

Voce	Descrizione
voice:en	Inglese
voice:en-us	Inglese Americano
voice:it	Italiano
voice:fr	Francese

Sono supportati i parametri `AudioServerSpeed` e `AudioServerVolume` in `hsyco.ini`.

Il valore di default del parametro `AudioServerSpeed` è 100: un numero più alto aumenta la velocità dell'annuncio vocale.

Il valore di default del parametro `AudioServerVolume` è 10, e deve essere un valore compreso tra 0 e 20, dove 20 genera un file audio con la massima ampiezza possibile.

Acapela

Se si utilizza Acapela, impostare il nome della voce tra quelle installate.

Se l'annuncio viene riprodotto attraverso telefoni SNOM, utilizzare le voci 8k e 22k.

Se invece viene riprodotto attraverso le telecamere Axis, utilizzare le voci 8k.

Mac OS X

HSYCO supporta il tex-to-speech a partire dalla versione Mac OS X 10.7 o successiva, utilizzando il motore tex-to-speech integrato. E' possibile installare voci aggiuntive attraverso il pannello delle preferenze di sistema di speech. I

nomi delle voci da dichiarare in HSYCO sono gli stessi mostrati all'interno del pannello.

Sono supportati i parametri `AudioServerSpeed` e `AudioServerVolume` in `hsyco.ini`.

Il parametro `AudioServerSpeed` è un numero positivo intero che rappresenta la velocità dell'annuncio vocale, espressa in parole al minuto.

Il parametro `AudioServerQuality` imposta il livello di qualità audio compreso nell'intervallo tra 0 (più basso) e 127 (più alto).

Il Formato dei File di Log

I file di log generati da HSYCO sono contenuti in sotto-directory di hsyco/logs.

HSYCO scrive un file di log distinto per ogni giorno, all'interno di una directory per ogni anno, nel formato **MMGG-message.log**. Ad esempio il file di log del giorno 12 Aprile 2012 sarà hsyco/logs/2012/0412-message.log.

Il dettaglio delle informazioni riportate nei file di log dipende dalle impostazioni della modalità di log definite in hsyco.ini. Normalmente sono comunque tracciati tutti gli eventi rilevanti rilevati sul campo, i comandi inviati, le informazioni di accesso degli utenti ed eventuali errori.

Ogni riga di log inizia con data e ora³³ nel formato:

AAAA.MM.GG HH:MM:SS:MMM

seguito dal carattere “+” in caso di errore, “-” in caso di messaggio informativo e “*” per i log remoti generati dal codice JavaScript dell'interfaccia Web.

Di seguito vengono riportati alcuni esempi di messaggi di log in diverse situazioni tipiche.

Avvio di HSYCO Senza Chiave di Licenza

```
2011.07.30 11:39:14.402 - HSYCO Ver. 2.10.0 (USER Ver. DEMO - build 12) started
2011.07.30 11:39:16.133 - Starting HTTP Server
2011.07.30 11:39:16.180 - Starting HTTPS SSL Server
2011.07.30 11:39:16.751 - HTTP server accepting connections on port: 80.
Document root: www
2011.07.30 11:39:18.126 - HTTPS server accepting connections on port: 443.
Document root: www
```

Quando la chiave di licenza software non è presente o non è valida, vengono attivati solo i server HTTP ed HTTPS.

³³ Viene utilizzata sempre l'ora locale in base alle impostazioni del server.

Avvio di HSYCO Normale

```
2012.01.04 11:39:14.402 - HSYCO Ver. 3.0.0 BETA Build 0086 (USER
Ver. No User Code) started
2012.01.04 08:14:10.943 - Starting HTTP Server
2012.01.04 08:14:10.951 - Starting HTTPS SSL Server
2012.01.04 08:14:11.167 - HTTP server accepting connections on
port: 80. Document root: www
2012.01.04 08:14:12.012 - Starting I/O Monitor [0]
2012.01.04 08:14:12.015 - Starting OpenWebNet Polling [0]
2012.01.04 08:14:12.019 - Starting Cameras Polling [0]
2012.01.04 08:14:12.021 - Starting Cameras Polling [1]
2012.01.04 08:14:12.022 - Cameras polling started [0]
2012.01.04 08:14:12.049 - Cameras polling started [1]
2012.01.04 08:14:12.078 - Starting User Time Callback Server
2012.01.04 08:14:12.110 - Starting Device Timer
2012.01.04 08:14:12.111 - User Time Callback Server started
2012.01.04 08:14:12.113 - Starting Timer Server
2012.01.04 08:14:12.115 - Device Timer started
2012.01.04 08:14:12.122 - Starting SysLog UDP Server
2012.01.04 08:14:12.124 - Timer Server started
2012.01.04 08:14:12.127 - Starting Music Players Status Polling
2012.01.04 08:14:12.134 - SysLog UDP server accepting packets on
port: 514
2012.01.04 08:14:12.138 - Music Players Polling started
2012.01.04 08:14:12.143 - Starting IRTrans Monitor [0]
2012.01.04 08:14:12.146 - Starting SlimServer Monitor
2012.01.04 08:14:12.151 - IRTrans Monitor [0] connection
established
2012.01.04 08:14:12.179 - SlimServer Monitor connection
established
2012.01.04 08:14:12.200 - HTTPS server accepting connections on
port: 443. Document root: www
2012.01.04 08:14:33.480 - ioMonitor - started [knx]
```

In condizioni normali, quando sono presenti anche telecamere, IRTrans, server Squeezebox Server e la centrale di controllo carichi sul bus SCS, vengono attivati tutti i sottosistemi di HSYCO:

- HTTP Server
- HTTPS Server
- I/O Server (per ogni sotto-sistema I/O)
- Cameras Polling (per ogni telecamera)
- User Time Callback Server

- Device Timer
- Timer Server
- Music Players Status Polling
- SysLog UDP Server
- IRTrans Monitor (per ogni IRTrans).

Per ciascuno di questi sistemi è normalmente presente una riga di log all'inizio dell'attivazione ed una a conferma dell'attivazione avvenuta.

Errori di Connessione all'I/O Server

```
2012.01.04 18:22:45.989 + ioMonitor - Exception [knx] - connect
timed out
2012.01.04 18:22:46.401 + ioMonitor - Handshake error (2) in
OpenWebNet monitor thread [myhome] - SCS server requires
authentication
```

In questo caso, viene segnalato un messaggio di errore di connessione per time-out al server KNX e un errore di autenticazione al gateway OpenWebNet.

Errori di Connessione a IRTrans

```
2012.01.04 18:22:46.916 + IRTransMonitor - Couldn't get I/O for
IRTrans [0] - Connection refused
```

Questo messaggio segnala un errore di connessione con il primo IRTrans – IRTrans [0] – definito nel parametro IRTrans in hsyco.ini.

Errori di Scrittura delle Immagini delle Telecamere

```
2012.01.04 20:00:00.000 + cameraRecSet - Camera recording skipped
- Disk full
```

Questo messaggio segnala l'impossibilità di salvare sul disco nuovi frame registrati dalle telecamere. Si verifica quando lo spazio disponibile sul disco risulta insufficiente.


```
2012.01.04 20:00:00.000 + cameraFlushAgedFrames - Early delete  
[camera] frames - Disk full
```

Questo messaggio segnala la cancellazione anticipata dei frame più vecchi di una telecamera. Si verifica quando lo spazio disponibile sul disco è limitato, ma non ancora insufficiente per la scrittura di nuove immagini.

Messaggi di Sicurezza

Tutti i messaggi relativi alla sicurezza sono identificati dall'intestazione:

SECURITY ALERT

PIN non trovato; del PIN errato sono riportate nel log solo la prima e l'ultima cifra:

```
2012.01.04 19:27:02.121 + SECURITY ALERT: CHECKAUTHREQ FAILED -  
REASON: PIN NOT FOUND - IP: 127.0.0.1 PIN: 1***5
```

PIN valido ma PUK errato; non viene trascritto il PIN ma la prima parte del valore crittografato del PIN stesso, identificabile nel file *access.ini*:

```
2012.01.04 19:30:44.739 + SECURITY ALERT: CHECKAUTHREQ FAILED -  
REASON: PIN/PUK AUTH FAILED - IP: 127.0.0.1 HPIN: 556342...
```

Errore nella URLKey presente nell'url di accesso all'interfaccia Web di HSYCO:

```
2012.01.04 19:29:47.111 + SECURITY ALERT: CHECKPINPAGEREQ FAILED -  
REASON: KEY NOT FOUND - IP: 127.0.0.1 KEY: alsldkekds
```

Il file *access.ini* è stato modificato e di conseguenza ricaricato da HSYCO:

```
2012.01.04 19:31:48.383 - SECURITY ALERT: access.ini file reloaded
```

Messaggi di login e logout dell'utente identificato dal nome *default*:


```
2012.01.04 19:28:16.734 - SECURITY ALERT: USER LOGIN - IP:
127.0.0.1 USER: default
2012.01.04 19:29:08.159 - SECURITY ALERT: USER LOGOUT - IP:
127.0.0.1 USER: default
```

Nuovo utente creato attraverso la pagina Web di gestione utenti:

```
2012.01.04 09:48:41.275 - SECURITY ALERT: WEB ADMIN - USER
[myuser] NEW [ack]
```

Modifica ad un utente esistente attraverso la pagina Web di gestione utenti:

```
2012.01.04 09:50:40.000 - SECURITY ALERT: WEB ADMIN - USER
[myuser] UPDATE [ack]
```

Cancellazione di un utente attraverso la pagina Web di gestione utenti:

```
2012.01.04 09:52:40.000 - SECURITY ALERT: WEB ADMIN - USER
[myuser] DELETE [ack]
```

Comandi Inviati dall'Interfaccia Web

Esempi di comandi inviati ai data point degli I/O Server, ai canali DMX e d esecuzione di codice Java personalizzato:

```
2012.01.04 20:04:25.956 - WEB I/O COMMAND [domino.o133.20=flip]:
[OK]
2012.01.04 21:44:18.493 - WEB I/O COMMAND [dummy.light.1=0]: [OK]
2012.01.04 21:44:31.252 - WEB I/O COMMAND [myhome.scene.
94.5=flip]: [OK]
2012.01.04 21:45:11.412 - WEB USER COMMAND [var1]: 2 [OK]
2012.01.04 21:47:21.300 - WEB DMX COMMAND [100*190]: [OK]
```


Localizzazione dei Client Web Wi-Fi

Quando per un Web Browser con una sessione HSYCO attiva viene rilevata l'informazione di posizione, cioè l'associazione ad un Access Point, il messaggio di log è:

```
2012.01.04 20:14:45.228 - LOCATION: 192.168.0.209 LOCATED AT 2p
```

Quando il client viene scollegato, cioè de-associato rispetto ad un Access Point, il messaggio è:

```
2012.01.04 20:16:17.752 - LOCATION: 192.168.0.180 POSITION UNKNOWN
```

Log delle Chiamate Telefoniche

Tutte le notifiche di chiamata gestite dalle centrali telefoniche integrate con HSYCO vengono registrate nel file di log.

```
2012.01.04 20:14:45.228 - PBX CALL [192.168.1.13]: FROM: 210 TO: 310 [OK]
```

Il File security.log

Qualora siano presenti delle centrali antifurto, verrà scritto il file **security.log**, nelle stesse directory dei file di log giornalieri. Questo file contiene l'elenco di tutti gli eventi di inserimento, disinserimento ed allarmi.

```
2012.01.04 12:38:25.861 [0] OFF - ZONES 1
2012.01.04 12:46:36.392 [0] ON - ZONES 1234
2012.01.04 12:47:08.703 [0] ALARM - INTRUSION - ZONE 1
2012.01.04 12:47:25.055 [0] OFF - ZONES 1234
2012.01.04 13:14:23.629 [0] ON - ZONES 1
```


Lista di Controllo Accessi

HSYCO supporta lato-server le liste per il controllo accessi per i comandi Web-based eseguiti dall'utente.

Il file `acl.ini` file è un file di testo che definisce le regole per consentire o rifiutare comandi.

Le modifiche al file `acl.ini` hanno effetto immediato.

Ogni riga definisce una regola di accesso, con il seguente formato:

action; user_id; location; type; command

Le regole vengono verificate a partire dalla prima riga e procedendo verso il basso fino a quando non viene trovata una corrispondenza in base all'utente, all'indirizzo IP del browser che ha inviato il comando, al tipo di comando e al comando stesso. Sulla regola che verifica la corrispondenza viene eseguita l'azione "allow" o "deny", consentendo o rifiutando l'esecuzione del comando.

E' possibile utilizzare il simbolo "*" nel campi `user_id`, `type` e `command` per trovare una corrispondenza con tutti i valori che iniziano o terminano con una stringa specifica.

La seguente tabella descrive tutti i campi di una regola.

Campo	Formato	Descrizione
action	allow deny	quando questa regola trova una corrispondenza, il comando viene eseguito o rifiutato
user_id	string [, string] * : tutti gli utenti	uno o più nomi utente (separati da virgole)
location	local remote *	"local" corrisponde a browser Web con indirizzi IP appartenente al gruppo degli indirizzi specificati in <code>trustedNet</code> ; "remote" a tutti gli altri
type	io timer camera * * : tutti i tipi	tipi di comando

Campo	Formato	Descrizione
command	string * : tutte le stringhe o sotto-stringhe	trova corrispondenza con la stringa di comando. i comandi "io" hanno il formato <datapoint>=<value> per gli I/O Server, dmx.<channel>=<value> per DMX, <IRTrans_id>=<command> per i comandi IRTrans, e <name>=<param> per i comandi utente. i comandi di tipo "camera" hanno il formato: <camera_name>=<function>.<action> per PTZ, o <camera_name>=<mail download> quando viene inviato un frame via e-mail o viene effettuato il download di un video registrato. i comandi "timer" hanno il formato: <timer_name>=<action>.

Se non viene trovata alcuna corrispondenza nel file acl.ini il comando viene rifiutato. Se è abilitato il log verbose, verrà mostrato un messaggio all'interno del file di log giornaliero:

```
2012.01.04 16:25:10.223 + ACL ALERT: DENIED - USER: staff
LOCATION: local TYPE: io COMMAND: user=test
```

Esempi:

```
deny; *; *; *; dmx*
allow; *; *; *; *
```

Abilita tutti i comandi da interfaccia Web, escluso il controllo DMX.

```
deny; *; *; camera; *zoom*wide
allow; *; *; camera; *
```

Disabilita tutti i comandi, consentendo di eseguire i soli comandi di controllo sulle telecamere, eccetto il comando di zoom.

```
deny; guest; *; *; *
allow; *; *; *; *
```

L'utente "guest" non è autorizzato ad eseguire alcun comando, mentre tutti gli altri utenti sono abilitati.

```
deny; *; remote; *; *
```

Impedisce l'esecuzione di comandi dai browser Web il cui indirizzo IP non sia compreso tra quelli definiti con il parametro trustedNet in hsyco.ini.

Certificati SSL per la Crittografia

HSYCO supporta la crittografia High-Grade a 256 bit (AES-256) per la protezione della comunicazione tra il Web Browser utilizzato dall'interfaccia Web ed il server HSYCO stesso. L'algoritmo AES-256 è considerato sicuro per applicazioni commerciali (ad esempio bancarie o di commercio elettronico) e per lo scambio di informazioni classificate³⁴.

Quando il certificato SSL (Secure Sockets Layer) è generato da HSYCO, quindi non firmato da una Certification Authority (CA) riconosciuta dai Web Browser disponibili sul mercato, è normale che il browser visualizzi un messaggio di sicurezza nel corso del primo accesso al sito o periodicamente all'inizio di una nuova sessione.

Questo messaggio serve a richiedere conferma all'utente circa la fiducia sull'identità del server verso il quale si sta iniziando una sessione sicura; infatti, non essendo il certificato firmato da una CA nota, il browser non può garantire autonomamente l'identità del server. Accettando il certificato la connessione viene stabilita e si può procedere normalmente.

In ogni caso, quindi anche se il certificato è generato da HSYCO, tutti i dati scambiati sono protetti comunque con la crittografia AES-256, allo stesso modo di quanto avviene con i certificati firmati.

Un certificato auto-generato garantisce lo stesso livello di protezione crittografica di un equivalente certificato firmato da una CA.

Il parametro **ServerName** nel file di configurazione **hsyco.ini** specifica il nome utilizzato per generare il certificato **SSL**, necessario per la crittografia del traffico Web HTTPS, e deve corrispondere al nome di dominio attraverso il quale il server HSYCO è accessibile via Internet.

³⁴ Nel Giugno 2003 il Governo USA ha certificato la possibilità di utilizzare gli algoritmi AES-256 anche per la protezione di informazioni classificate a livello TOP SECRET.

Il certificato è contenuto nel file **hsyco.keys**. Quando HSYCO viene avviato, se questo file non è presente, viene generato automaticamente un nuovo certificato SSL in base al nome definito in ServerName. Altrimenti HSYCO si limita a utilizzare il certificato contenuto in questo file, che può quindi anche essere stato generato da una **CA** ufficiale. A meno che non si sia scelto di utilizzare un certificato ufficiale, questo file viene quindi creato e gestito da HSYCO senza bisogno di interventi manuali.

Se il nome definito in ServerName viene modificato, HSYCO creerà automaticamente un nuovo file hsyco.keys al successivo riavvio.

E' comunque possibile inserire nel file hsyco.keys un certificato firmato da una CA. La procedura, che può essere eseguita con comandi di console grazie al software di gestione dei certificati presente in HSYCO SERVER, è descritta di seguito, utilizzando a titolo di esempio la Certification Authority StartCom Ltd³⁵.

Accedere alla console via SSH:

```
ssh hsyco@192.168.0.50
```

ed eseguire i comandi seguenti nella directory principale di HSYCO, in cui si trova il file hsyco.keys, o meglio in una directory temporanea, dalla quale alla fine della procedura sarà sufficiente copiare hsyco.keys nella directory principale di HSYCO.

Questi comandi richiedono una password per la protezione del file hsyco.keys e dei certificati in esso contenuti. Tuttavia, poiché HSYCO deve essere in grado di accedere a questo file automaticamente, la password deve essere la stessa utilizzata da HSYCO per l'accesso in lettura al file. Utilizzare sempre la password:

```
hsycopass
```

³⁵ Start Commercial (StartCom) Ltd. - <http://www.startssl.com/>

nei comandi seguenti. Poiché la password utilizzata non è segreta, è importante ricordare che il certificato firmato dalla CA sarà potenzialmente utilizzabile da chiunque dovesse entrarne in possesso. La protezione dell'accesso alla console di HSYCO, utilizzando una password di accesso per l'utente hsyco di adeguata lunghezza e complessità, diventa per questo motivo ancora più importante.

1) generare la chiave per il nome di dominio:

```
keytool -genkey -keyalg rsa -alias hsyco -dname "cn=www.domain.com,
o=domain, o=.com" -keystore hsyco.keys
```

2) creare il file di richiesta del certificato:

```
keytool -certreq -alias hsyco -file www.domain.com.csr -keystore hsyco.keys
```

3) importare le chiavi della CA scelta nel file hsyco.keys:

```
wget http://www.startssl.com/certs/ca.crt
keytool -import -alias startcom.ca -file ca.crt -keystore hsyco.keys
wget http://www.startssl.com/certs/sub.class1.server.ca.crt
keytool -import -alias startcom.ca.sub -file sub.class1.server.ca.crt -
keystore hsyco.keys
```

4) ottenuto il certificato firmato dalla CA, trasferire il contenuto in un file, ad esempio:

```
cat > www.domain.com.signed.crt
```

5) infine importare il certificato nel file hsyco.keys ed eventualmente trasferire questo file nella directory principale di HSYCO:

```
keytool -import -alias hsyco -file www.domain.com.signed.crt -keystore
hsyco.keys
```


Database SQL

HSYCO integra HSQLDB³⁶ (HyperSQL Database), un sistema per la gestione di basi di dati relazionali al quale le applicazioni possono collegarsi tramite le API del driver JDBC³⁷.

HSQLDB offre un motore di database transazionale di piccole dimensioni e una veloce gestione dei processi multithread mettendo a disposizione tabelle in-memory o disk-based. Inoltre supporta modalità server e embedded.

HSQLDB è direttamente integrato all'interno del core di HSYCO. Non è necessario aggiungere alcun file jar al classpath di HSYCO.

Il database utilizzato da HSYCO si chiama “hsyco” e i file sono salvati nella directory data/. La directory data/ è da considerarsi come un percorso riservato ed è pertanto sconsigliabile utilizzarla per memorizzare database dell'utente o, in generale, per altri scopi.

HSYCO ricorre all'HSQLDB per memorizzare variabili persistenti.

In futuro esso potrà essere utilizzato per altri scopi, perciò è importante seguire alcune semplici linee guida in modo da evitare potenziali conflitti con le future release.

E' importante sottolineare che Il carico di risorse del motore SQL influisce direttamente sulle performance e la stabilità di HSYCO SERVER.

³⁶ HSQLDB (HyperSQL DataBase) è uno dei principali motori di database SQL relazionali scritti in Java. HSQLDB possiede un driver JDBC e supporta quasi completamente l'ANSI-92 SQL più numerosi aggiornamenti di SQL:2008. L'intera documentazione HSQLDB è disponibile all'indirizzo: <http://hsqldb.org/>.

³⁷ JDBC (Java DataBase Connectivity), è una interfaccia per database che consente l'accesso alle basi di dati da qualsiasi programma scritto con il linguaggio di programmazione Java, indipendentemente dal tipo di DBMS utilizzato. È costituita da una API, raggruppata nel package java.sql, che serve ai client per connettersi a un database. La documentazione JDBC è disponibile all'indirizzo: <http://java.sun.com/products/jdbc/overview.html>.

Connessione

Prima di tutto è necessario impostare gli import:

```
import java.sql.*;
```

Viene effettuata la connessione al database chiamato *mydb* nella directory *userdata/*, con un ritardo di scrittura di 100 millisecondi.

E' possibile utilizzare diversi parametri di connessione:

```
static Connection dbConnection =  
DriverManager.getConnection("jdbc:hsqldb:file:userdata/  
mydb;hsqldb.write_delay_millis=100", "user", "password");  
dbConnection.setAutoCommit(true);
```

Normalmente dovrebbe essere impostata la modalità *autocommit* se non è necessaria la gestione delle transazioni.

Disconnessione

```
try {  
    dbConnection.createStatement().execute("shutdown");  
    variablesDatabaseConnection.close();  
} catch (Exception e) {}
```

Si tratta del tradizionale comando di arresto, cui segue una chiamata che chiude la connessione del database. Questa sequenza deve essere utilizzata se non si prevede di accedere nuovamente al database nel breve periodo.

Esiste un'ulteriore modalità di chiusura del database, il comando *SHUTDOWN COMPACT*. Questo processo deve essere periodicamente eseguito per ripulire e minimizzare le dimensioni dei file.

Tabelle

HyperSQL supporta la definizione di tabelle persistenti e ne definisce tre tipologie a seconda della modalità di memorizzazione dei dati: le tabelle *memory*, *cached* e *text*.

Le tabelle *memory* costituiscono la tipologia di default quando si utilizza il comando *CREATE TABLE*. Si tratta di tabelle persistenti ma il salvataggio dei dati nei file e la riletture diventano molto onerosi in termini di tempo, soprattutto se il database è di grandi dimensioni.

Le tabelle *cached* vengono create attraverso il comando *CREATE CACHED TABLE*. Solo parte dei loro dati e indici rimangono in memoria. Questa caratteristica è fondamentale nella gestione di tabelle di grandi dimensioni, in quanto il motore di database impiega meno tempo per effettuare l'avvio. Il loro svantaggio consiste in una riduzione della velocità. Si sconsiglia di utilizzare tabelle *cached* se la base di dati è relativamente piccola.

L'esempio seguente mostra come creare una tabella:

```
try {  
    dbConnection.createStatement().execute("create table  
mytable (id varchar(255) not null unique, value  
varchar(65536))");  
} catch (Exception e) {}
```

Operazioni sulle Tabelle

Per eseguire operazioni sulle tabelle, come query, inserimenti e aggiornamenti, possono essere utilizzati due tipi di classi: *Statement* e *PreparedStatement*.

Una classe di tipo *java.sql.Statement* viene utilizzata per eseguire query e statements. Essa può essere riutilizzato per eseguire ogni volta uno statement diverso.

Una classe di tipo *java.sql.PreparedStatement* consente di eseguire ripetutamente un singolo statement, solitamente composto da parametri che assumono nuovi valori prima di essere riutilizzati. Quando viene creato un oggetto *PreparedStatement*, il motore mantiene compilato il codice SQL in caso di riutilizzo, finché l'oggetto non viene chiuso.

Di conseguenza, è più veloce l'uso ripetuto di una classe *PreparedStatement* rispetto a *Statement*.

Di seguito vengono mostrati alcuni esempi di utilizzo per effettuare una query, un inserimento e un aggiornamento di una tabella del database.

Query

```
try {  
    PreparedStatement stmt =  
dbConnection.prepareStatement("select value from mytable  
where id = ?");  
    stmt.setString(1, id);  
    ResultSet rs = stmt.executeQuery();  
    if (rs.next()) {  
        String value = rs.getString(1);  
        rs.close();  
        return value;  
    } else {  
        return null;  
    }  
} catch (Exception e) {}
```


Inserimento

```
try {  
    PreparedStatement stmt =  
dbConnection.prepareStatement("insert into mytable values  
(?, ?)");  
    stmt.setString(1, id);  
    stmt.setString(2, value);  
    stmt.execute();  
} catch (Exception e) {}
```

Aggiornamento

```
try {  
    PreparedStatement stmt =  
dbConnection.prepareStatement("update mytable set value=?  
where id=?");  
    stmt.setString(2, id);  
    stmt.setString(1, value);  
    stmt.execute();  
} catch (Exception e) {}
```


Il File access.ini

Questo file contiene le informazioni relative ai **PIN/PUK** di accesso degli utenti. Il file può essere modificato manualmente per creare o cancellare PIN, ma normalmente viene modificato attraverso le pagine di gestione utenti nell'interfaccia Web di HSYCO. Viene anche aggiornato da HSYCO in seguito ad ogni accesso, con l'orario e l'indirizzo IP dell'ultimo accesso ed il numero di errori di accesso per ciascun utente.

E' possibile disabilitare temporaneamente l'accesso per un utente e definire per ogni utente i diritti di accesso alle sotto-directory www dell'interfaccia Web.

Le modifiche effettuate manualmente al file access.ini vengono ricaricate automaticamente da HSYCO entro 15 secondi dal salvataggio del file, non è quindi necessario un riavvio di HSYCO per rendere efficaci le modifiche.

Ogni riga del file identifica un utente autorizzato ad accedere a HSYCO. Il formato è:

nome_utente = parametri

Non ci sono limiti al numero massimo di utenti. Ogni utente deve avere un nome identificativo unico e un PIN distinto. Il PIN è un codice esclusivamente numerico (0-9) di 5 cifre. Creare due utenti diversi con lo stesso PIN è un errore di configurazione, che HSYCO risolve ignorando gli utenti con PIN duplicato.

nome_utente può essere preceduto dal carattere “*” per indicare un utente con diritti di amministratore³⁸.

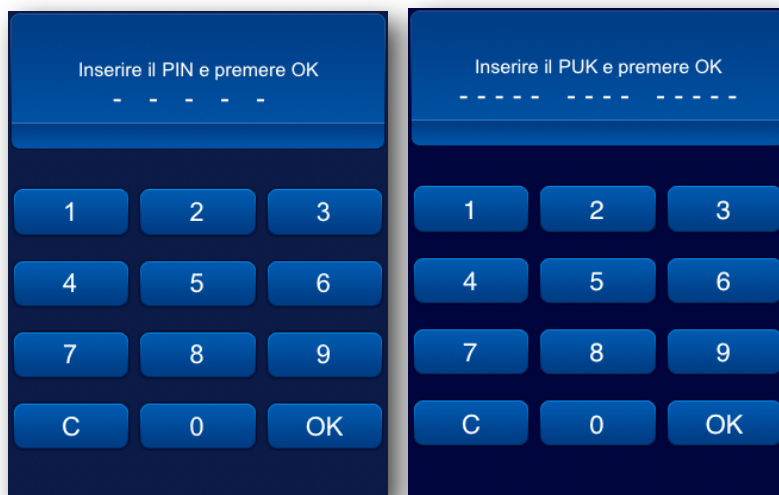
E' possibile creare un nuovo file access.ini, o aggiungere manualmente un utente ad un file esistente, indicando semplicemente il nome dell'utente ed il PIN in chiaro, ad esempio:

³⁸ I diritti di amministratore consentono in particolare di creare, modificare e gestire i diritti di accesso di altri utenti dell'interfaccia Web di HSYCO.


```
*user=00000
```

Dopo aver salvato il file access.ini con i nuovi utenti è possibile completare la registrazione di questi utenti accedendo direttamente all'interfaccia Web:

<https://192.168.0.50/small/pin.hsycoserver>



Introdurre uno dei PIN definiti e premere [OK], quindi alla richiesta del PUK introdurre un codice di 14 cifre e premere [OK]. Il codice inserito diventa il PUK dell'utente associato al PIN e viene memorizzato da HSYCO insieme al PIN in modo non più leggibile nel file access.ini.

Per motivi di sicurezza e riservatezza i codici PIN e PUK sono memorizzati in modo non leggibile; non è normalmente possibile recuperare i codici associati ad un utente. E' quindi opportuno trascrivere questi codici in un posto sicuro.

Per garantire la massima sicurezza è importante che il file access.ini sia trattato come un documento riservato³⁹, evitando la copia o l'invio del file su sistemi non protetti e l'accesso a persone non autorizzate.

In qualunque momento è possibile revocare l'accesso ad un utente semplicemente eliminando dal file access.ini l'intera riga di definizione dei parametri utente, senza che sia necessario riavviare HSYCO. L'accesso viene

³⁹ Pur essendo il contenuto di access.ini in parte crittografato, il recupero di PIN e PUK in chiaro è comunque possibile utilizzando particolari tecniche.

revocato entro pochi secondi dalla modifica di access.ini anche per le eventuali sessioni attive in corso.

Per disabilitare temporaneamente l'accesso per un utente, senza dover cancellare l'intera riga dei parametri, è sufficiente inserire il carattere “#” immediatamente dopo *nome_utente*=, come nel seguente esempio:

```
#Generated by HSYCO 3.0.0
#Wed Jan 04 14:25:29 CET 2012
user=#55634e926185944768b4f3,d29995f54ef9b1b169e7c,
1230989129162,0,192.168.0.180
```

Per limitare l'accesso di un utente ad una o più sotto-directory *www* dell'interfaccia Web, è possibile specificare i nomi delle sotto-directory autorizzate, separate da spazi, nella riga dei parametri dell'utente subito dopo la virgola del secondo parametro:

```
user=55634e9261b4f3,d29995f169e7c,garage garden,1230989129162,0,192.168.0.180
```

In questo esempio il PIN associato all'utente *user* potrà accedere solo alle interfacce Web definite nelle sotto-directory *www/garage* e *www/garden*.

E' possibile specificare un massimo di tre sotto-directory distinte per ogni utente.

Release Notes

3.1.2

server updates:

- Mitsubishi MNET I/O Server: performance improvements
- HSYCOREMOTE I/O Server: some data-point change events were not propagated correctly to the supervising HSYCO server
- new URL format for PBX calls notifications: `/x/vcall?from=<from>&to=<to>`
The old format, `/x/vcall/<from>/<to>` continues to be valid too
- the HTTP camera recording trigger API now also supports the following syntax: `/x/camerarec?camera=<name>&zone=<id>`
- new `CommPortsList` option in `hsyco.ini`, allows explicit definition of operating system's local serial ports

bug fixes:

- minor fix to the EVENTS language parser for the IO action
- scheduler time bug (on new, it showed "00:" on time fields)
- camerapanel motion popup showed up in wrong position when in a popup page
- the camera client is now more tolerant to HTTP content where lines are not always terminated with CR LF (as seen with GANZ cameras)
- Domino I/O Server: the (temp) object's DFCT configuration function was broken in HSYCO 3.1.0
- Domino I/O Server: energy metering data point were not updated
- manual clock update admin page wasn't working
- fixes a compatibility problem with latest versions of Firefox and Opera, that affected the select objects in admin pages
- fixes a bug in the Web page reload process, that could cause performance problems on HSYCO updates or www files changes

3.1.1

server updates:

- Modbus I/O Server: writing readholdingregisters:bitshort, readholdingregisters:bitint, readholdingregisters:bitlong, readinputregisters:bitshort, readinputregisters:bitint and readinputregisters:bitlong, it is now possible to read 1, 2 or 4 registers returning the individual bit values
- Domino I/O Server: added support for DF4RP/I and DFDT

EVENTS

- the update of variables is now executed atomically for the list of actions associated to an event condition. For example, time : \$v + 1, \$v + 1 used to trigger two distinct variable change events, because \$v was changed twice on the time event. Now the same action still causes \$v to be increased by two every minute, but only a single \$v event is triggered, after both actions have been executed

GUI updates:

- improved browser performance when showing data loggers
- (scheduler) object: time from and time to now show 00:00 if not set, instead of the current time

bug fixes:

- (scheduler) object error on some dates; entering dates of the month of December, and few other days in other month, caused a server-side validation error that prevented the scheduler from saving the change
- fixes a login issue in the Manager that caused all login requests after a failed one to fail again until the Manager page was manually reloaded
- (csx75) object of the CSX75 I/O Server: the PIN entry procedure was not working properly in some cases
- GSM I/O Server: the I/O Server occasionally restarted after transient error responses from the modem
- changes to the MJPEG streaming to address a problem caused by the latest versions of Safari browser
- WXONLINE I/O Server: minor fixes
- fixes to the (dmx) object
- minor fixes to the dimmer slider control

3.1.0

server updates:

- new BACnet I/O Server
- new HSYCO I/O Server: remote, hierarchical access to HSYCO's data-points
- new scheduler server
- audio support: text to speech and files playback to the server's line-out, multicast to SNOM phones, Web clients, Axis cameras and Rovio
- new Interlogix FP2000 I/O Server
- new EL.MO. security panels I/O Server
- new ATON AH66 I/O Server: multi-room audio system
- new GSM I/O Server: send GSM messages using any standard AT-protocol serial GSM modem
- added support for the ENTTEC DMX USB PRO DMX gateway
- DOMINO I/O Server: added support for the DFANA energy meter module
- Guardall I/O Server: added support for the PX80 panel, and changed log with zone names instead of numbers
- Modbus I/O Server: new error events on Modbus read or write errors
- Modbus I/O Server: new option extendedNames. When this option is set, data point names used to return values will have the postfix .FF, where FF is the 2-digit decimal function code
- WXONLINE I/O Server: added support for Weather Underground and Yahoo data sources
- integrated network time synchronization and time-zone configuration (TimeZone and TimeAutoUpdate params in hsyco.ini)
- improved database disk usage
- changes to the network configuration setting procedure

EVENTS:

- new AUDIO action for text-to-speech and audio files play
- string concatenation support on values in COMM, IO and USER actions
- the CAMERA event now supports the optional zone information: CAMERA name [= zone_id]
- assignment of an empty string "" to a variable is now allowed. Events like \$var="" match both if \$var doesn't exist and if \$var was explicitly assigned an empty string.
- \$var = IO datapoint where datapoint doesn't exist also sets \$var to the empty string, while it didn't change the variable's value in previous versions
- external URL support for (background) and (image) UISET img attribute
- the MAIL action can now send regular files as attachments, and supports sending messages via a authenticated email accounts, also with SSL

- the DATALOGGER name = FILE ... actions now accept multiple strings, including variables, as the file name
- Leak Detector:
 - commands: LEAK <name> = <value>, LEAK <name> = PERIOD <seconds>, LEAK <name> = DEVIATION <percent>, , LEAK <name> = CLEAR
 - event: LEAK <name> = on|off

Java API:

- the sendMail() method can now send regular files as attachments, and supports sending messages via a authenticated email accounts, also with SSL
- Leak Detector APIs: LeakDetectorClear(), LeakDetectorOptions() and LeakDetectorUpdate()
- new audioPlay() methods for text-to-speech and audio files play

GUI updates:

- new (scheduler) object
- new (selector <file>; <pos>; <width>; <height>; <container id>; <group>; <text>; <css>) object
- new (submitimage) object
- new (bacnetbrowser) object
- new UISET object's "blink" attribute
- (panel) now has a default transparent color
- (container) now has an optional attribute to set the container's default visibility
- left CSS style attribute added for text fields in (buttonimage), (image), (imagelink), (text), (userimage)
- (datalogger) now has additional options for the legend format. The new legend attribute can be set to "false", "true", or any combination of "min", "avg" and "max". The totals attribute can be set to "true" to show summary values, or "false" to only show text labels in the legend
- (image), (imagelink), (userimage): text can now be set with UISET

Manager / Project Editor:

- project editor:
- the video object now has a video file selection pop-up
- the image and video file selection pop-ups now default to www/img
- choose file pop-up now remembers last opened folder
- tabs in project panel renamed to select, edit and +

general:

- improved server restart detection and automatic page reload on hsyco.jar changes

bug fixes:

- High Availability: the data directory should not be synced from master to slave
- MAIL action and sendMail() method: sending a mail with an embedded reference to not-existing or disabled camera caused an internal error preventing the whole message from being sent
- improved UTF-8 character set support
- URL bar in Android browser is now properly removed
- Manager: the log viewer text can now be selected for copy&paste when paused
- Data Loggers: fixed handling of erroneous values passed to a counter data logger
- fixes to the license limits validation logic

3.0.3

Core:

- CONTATTO and DOMINO I/O Servers: optimized performance of registers status polling
- minor changes to the Guardall QX32i I/O Server
- DataLogger: refresh is now automatic; refresh is automatically performed at the beginning of each hour. The explicit refresh action is now obsolete
- new I/O Server for the Aritech CSx75 (Comfort) security panels
- variables can now be used in value strings

EVENTS:

- new URL actions and events with HTTP/HTTPS support
- new \$DATE:DOW\$ predefined variable returns a number that represents the current day of week
- new *= string contains operator. The string1 *= string2 condition is true if string2 is contained in string1. This operator is case-insensitive
- concatenated strings are not supported as values passed to the UISET action

index.hsm / GUI:

- new (datalogger <logger id>; <pos>; <width>; <height>; <val label>; <attributes>) object
- added label text with CSS attributes to (image), (imagelink), (buttonimage) and (userimage)

Manager:

- using a PC with a physical keyboard, you can press [Ctrl] and [Shift] at the same time for fast switching between applications
- Log Viewer controls added to File Manager mini-logs
- Log Viewer's log lines buffer increased to 1000 lines
- Log Viewer's new pause button
- a confirmation pop-up is shown if the user tries to leave the Manager's web page while uploading a file or changes have not been saved in the Project's Editor or File Editor
- when editing events.txt, a new warning is shown if the file contains errors, also listing all errors
- when events.txt is uploaded using the File Manager, reloading occurs immediately
- support for UTF-8 characters in the File Manager's text editor has been improved

Bug-fix:

- improved page switching, to prevent pages overlapping effect on very slow browsers
- backgrounds in manager appear in the same position as they do on the client (top: -5px)
- CONTATTO I/O Server: the server failed to initialize when an unsupported device was configured in the MCP
- saving a file with the File Manager and immediately shutting off the server could cause the loss of data. Risk of data loss is now significantly reduced
- fixes to the Modbus write functions
- minor fixes to the Data Logger engine
- minor fixes to (chart) object
- minor fixes to the EVENTS interpreter

3.0.2

Core:

- new I/O Server for the Guardall QX32i security system

Bug-fix:

- back broken on (music) and (timer)
- keys malfunction in timer panels on iOS and Android

3.0.1

Core:

- DOMINO and CONTATTO: you can now limit the number of virtual points and registers read and written by HSYCO, and have access to all of the 1024 registers; defaults are unchanged
- DOMINO: added support for DF8IL, DFDI2B and DFRHT modules
- Data Loggers: if you want a Data Logger to set the discrete UI objects, you should now set `DataLoggers.<id>.UseCharts = true`
- the ACL system now allows you to define rules based on the client's IP address being inside the trusted range (local) or outside (remote)

EVENTS:

- if one of the two operands is a number and the other is "off", replace the latter with 0 before comparing
- new ROUND operator for rounding numeric values in variables to an arbitrary number of decimal digits (from 0 to 9). E.g.: `$MYVAR ROUND 2`
- new FORMAT operator formats numeric values in variables based on a format pattern. E.g.: `$MYVAR FORMAT #####.00`
- new POWER transient event (so you can set an event that reacts to any change in power, without the condition)

Java API:

- `PowerEvent()` changed to `public static int PowerEvent(int power)`

index.hsm / GUI:

- pages have a new "protected" parameter that requires the user to login again when accessing the page
- new "gauge" type for the (chart) object
- (chart) object default colors changed: background is now transparent, and labels are white
- new `(keypadlid <pos>; <min>; <max>; <digits>; <decimals>; <type>; <label>; <css>)` object
- we now have a local graphic effect when a button is pressed

Manager:

- about popup with version information
- when `hsyco.ini` is changed, it reloads the configuration after the server has restarted
- new HSYCO restart button when the `hsyco.jar` file is selected
- search or exclude now works with multiple words in the search field, matching any word you have entered

- when systemtopo.txt is updated, objects' graphics change (e.g. button appearance changes from LIGHT to AUTOM)
- changes to the new project panel

Bug-fix:

- I/O Servers HWGIO, HWGMONITOR, EKAHAU, HIDEEDGESOLO, SAMSUNG VIDEO ANALYSIS and SAMSUNG CAMERA I/O could block the update of the systemtopo.txt file at start-up
- Tridonic/DALI: dimmer values were not returned with ending % character
- (adminlink) objects failed to show the admin pages
- Tecnoalarm driver:
 - in the high-availability configuration, the not-active server should not connect to the Tecnoalarm units
 - in some cases, a communication problem could force the driver to erroneously reload all past logs
- (buttonimage) caused an error if some parameters were not defined; now they are optional
- access.ini and other system's files are written to disk using a more robust procedure
- light_rain icon for (weather) was missing
- (cameralink) not working with grids and the default destination panel id
- minor fixes to the Data Logger
- various fixes to prevent the manager to stop working when the server is offline (like switching from file manager to project editor or saving a project)
- reversed value series for (chart) objects with horizontal orientation
- DOMINO: fix to the (temp) object's set-point programming function
- fixes a problem playing recorded video on Firefox with MJPEG streaming mode enabled
- fixes to the (nuvo) object

3.0.0

Core:

- support for DUEMMEGI's DOMINO home automation system
- support for DUEMMEGI's CONTATTO home automation system
- support for NuVo's Grand Concerto and Essentia E6G multi-room system
- support for FCC Planterm RayCONTROL
- support for YAMAHA RX-Vxx67 AV Receivers/Amplifiers
- new AXISDEC I/O Server for control and streaming to the AXIS P7701 Video Decoder
- extended license manager to support additional license limits options (CAM, DMX, SEC, IOS, COM, IRT, SQU, DOMINO, CONTATTO, KNX)

- support for TCP/IP serial servers with server failover option, including the Advantech EKI serial port IP gateway; configuration parametri:
 CommPort.<id>.Type = SERVER
 CommPort.<id>.IP = <IP address or server name> [, <failover server IP address or name>]
 CommPort.<id>.Port = <port number>
- support for PowerOne inverters
- support for Mitsubishi M-NET gateways
- support for KNX BAOS Object Server gateway
- enhanced Project Editor
- new File Manager
- new Log Viewer
- support for license key auto-provisioning
- support for server-side access control lists for Web-based user commands. The optional acl.ini file defines the rules to allow or reject commands
- new dummy I/O Driver. It's a fake driver that could be useful for projects development and demonstrations. Supports lighting and automation
- new URL query option "nocache" disables the HTML5 persistent cache for the Web session, overriding the default settings
- support for Tridonic DALI SCi2 RS-232 DALI gateway
- support for Carlo Gavazzi ISMG inverters
- redesigned My Home I/O Server
- new WXONLINE I/O Server retrieves local weather data from Google
- in hsyco.ini, setting trustedNet = local or not defining this parameter, HSYCO will consider all local network addresses as trusted
- HTTPS call to /x/hsyco.pem returns the self-generated root certificate

EVENTS:

- new PING event and action. A "PING <hostname1> [<hostnameN> [=<timeout>]]" action tests the reachability of one or more hosts, identified by their hostnames or IP addresses, within the optional timeout defined in milliseconds (or using a default timeout of 200 ms). This test generates PING events like "PING <hostname> = ON" if the host is reachable, or "PING <hostname> = OFF" if not reachable
- you can now assign the current value of I/O and DMX devices to variables and UI objects , for example: \$myvar = IO gate.o22.1, UISET uiobject.text = DMX 33
- new DATALOGGER action to easily create counters and range historical charts (hourly, daily, monthly charts for present and past periods)
- new Modbus actions and events: it is now possible to trigger the Modbus read functions on arbitrary addresses, with I/O events generated when data is read:

- actions:

- IO <modbus IO gwid>.<unit id>.<register address> = READCOILS:<quantity of coils>
- IO <modbus IO gwid>.<unit id>.<register address> = READDISCRETEINPUTS:<quantity of inputs>
- IO <modbus IO gwid>.<unit id>.<register address> = READINPUTREGISTERS:<USHORT | UINT | ULONG | SHORT | INT | LONG | FLOAT>
- IO <modbus IO gwid>.<unit id>.<register address> = READHOLDINGREGISTERS:<USHORT | UINT | ULONG | SHORT | INT | LONG | FLOAT>

- events:

- for READCOILS and READDISCRETEINPUTS: IO <modbus IO gwid>.<unit id>.<register address>.<coil/input number> = <0 | 1>
- for READINPUTREGISTERS and READHOLDINGREGISTERS: IO <modbus IO gwid>.<unit id>.<register address> = <value>
- added support for Modbus write actions (single coil, single/multiple registers with short, int, long and float numbers)
- new POWER action, equivalent to the powerSet() method, sets the electric power state

Java API:

- if you force a page or pop-up display by returning a page:... string on the userCommand() callback, this method will be called again when that pop-up or page is closed, with "/close" appended to param
- enhanced uiSet(id, attr, value):
 - id = * is a broadcast to all open projects. Only supported for attr = page, attr = lock and attr=pageback
 - when used with attr=page, will force all browsers that have any index.hsm visible, to switch to the specified page id, or menu. Setting value to an empty string will automatically revert to the previous page
 - when used with attr=lock, will lock or unlock the navigation of all clients.
 - when used with attr=pageback, and value=<pageid>|<popupid> will force all clients that are currently displaying page <pageid> back to the previous page, or clients showing the pop-up <popupid> to close that pop-up
 - note that the pageback function can also be used with id=<projectid>, not only for all (*) projects, to control a specific project
- new user.user(String name, String param) command method, triggers userCommand() callback and USER events
- new dataLoggerInit(), dataLoggerRefresh(), dataLoggerUpdate(), dataLoggerOptions(), dataLoggerSave() and dataLoggerClear() user command methods for data charting

- new static byte[] modbusWriteMultipleRegisters(String name, int unit, int address, byte[] mask, byte[] bytes) method
- new static byte[] modbusReadCoils(String name, int unit, int address, int quantity) method
- new static byte[] modbusReadDiscreteInputs(String name, int unit, int address, int quantity) method
- new static byte[] writeSingleCoil(String name, int unit, int address, boolean value)
- new static byte[] writeSingleRegister(String name, int unit, int address, byte[] bytes)
- new static void closeComm(String portName) method. Closes a serial port connection when the serial port is defined as a “server” type, and is ignored otherwise. When used with serial servers having a fail-over configuration, this method forces a reconnect at the next call of readComm() or writeComm(), so that a switch between the fail-over IPs would happen if the current IP is not responding
- the LocationEvent() callback is called with the IP parameter set to null if the associated device is not connected to the HSYCO Web interface
- haActiveState() replaces haActiveState(int channel)

index.hsm / GUI:

- classicbig skin renamed to blue
- orientation mode (#size <landscape size>[; <portrait size>])
- pop-ups are centered when they don't fit in the page
- compact mode for interfaces with a width less than 500:
 - icons behave like previous small skins
 - icons with no text
 - bottom bar: location, power
- new (sliderbutton) object
- www/admin/index.hsm is now embedded in the hsyco.jar distribution file
- live video from cameras is now delivered as MJPEG streams on Safari, Firefox and Chrome browsers. The ?nocamstream URL option disables MJPEG streaming
- when the HTML5 persistent cache is enabled, the cache is reloaded in the background before reloading the page
- support for camera recordings video files download
- new (temp) and (tempmini) objects for zone temperature control
- new (#locked true) directive, to disable any user command from the GUI of a project
- the (background) object can be assigned an id, and you can change the img attribute dynamically
- chart object have an attributes parameter for initialization ([name]=[value]; ...)

- removed the status message field parameter from the (user), (usermini), (usermicro), (userimage), (userrgb) objects
- removed the status message field parameter from the (ir), (irmini), (irmicro) objects
- (charts) object support for empty values (ex. values=1,2,,4)
- (charts) object new "origin" parameter, moves the origin to a value different from 0 (ex. origin=3.2)
- new (buttonimage) object
- new (weather) object
- if the (submitid) object's id starts with \$, then the server will automatically set variables named \$<id>.<input id> for each input field to the values in the input fields

Bug-fix:

- EVENTS: actions like IO <id1> = IO <id2> did not work
- fixed PIN/PUK keypad problem with Android clients
- the Scheduler Server could hang in a time-out state when the system clock is changed
- chart fix
- devices added to the systemtopo.txt files are sorted alphabetically
- public www files are not listed in the manifest file and are not checked for www files changes

2.10.4

Bug-fix:

- Serial ports driver: bug fixes for the TCP/IP driver
- fixed bug in project editor: address for music objects was not initialized
- My Home driver fix for temperature control with multiple bus gateways and extended addressing
- in systemtopo.txt, the backslash character, single and double quotes are properly escaped to prevent client-side errors
- Tecnoalarm driver: in the high-availability configuration, the not-active server should not connect to the Tecnoalarm units

2.10.3

Bug-fix:

- fixed bug in project editor: address for music objects was not initialized

2.10.2

Bug-fix:

- fixed a UI problem with pop-ups on Microsoft Internet Explorer

2.10.1

Core:

- HID I/O Driver:
added support for event 2021: Granted Access - Extended Time, with event data-point <servername>.access.granted.extended

2.10.0

Core:

- support for BTicino My Home “Advanced Addressing”. You can set the openServersAddressMode = advanced parameter in hsyco.ini to enable this feature. When using multiple buses the 4 least significant digits of an address will represent the AAPP device address, the second bus will have addresses starting at 10000, the third start with 20000 etc.
- HSYCO can optionally serve live images of its cameras or grids through a password protected HTTP request, like `https:<hsycoserver>/x/camera/<cameraid>?password=<password>&size=<width>x<height>` (size is optional) for single frames or `https://<hsycoserver>/x/camerastream/<cameraid>?size=<width>x<height>&password=<pwd>[&period=<millis>]` to retrieve a MJPEG stream. This feature is enabled setting the Camera.<id>.RemoteRequestPassword parametri in hsyco.ini.
To enable the same feature on grids, use “grid<N>” as the <cameraid> in the above URLs, and set the CameraGrid.<N>.RemoteRequestPassword parametri in hsyco.ini. It is also possible to have different passwords for clients within the trusted range of IP addresses and outside.
Use the Camera.<cameraid>.TrustedRequestPassword and CameraGrid.<N>.TrustedRequestPassword parametri, setting passwords that must be different from RemoteRequestPassword, or will be ignored. Note that requests with the TrustedRequestPassword value will be also served via HTTP, without SSL.
- support camera frames acquisition using HTTPS with self-signed certificates
- support camera frames acquisition through MJPEG streams
- support for SAMSUNG IP cameras, including motion detection, I/O, video analysis and PTZ for some models. Two new parametri in hsyco.ini are used to define the camera model type, Camera.<camid>.Type, and to optionally

enable I/O and VA, Camera.<camid>.IO=enabled. If you want to enable I/O only, set it to “enabled:io”, or to “enabled:va” to only enable VA

- support for HTML 5 off-line cache storage with Google Chrome 8.x browsers
- improved start-up time of the HSYCO server. Now the HTTPS server starts almost immediately. The motion markers rebuilding process was redesigned as a thread and runs in parallel at start-up. All threads start immediately, while cameras recording is disabled until the markers rebuild is completed
- new HTTPServerPublicDirectory in hsyco.ini, enables a basic Web server that serves files, without any parsing, contained in a specific directory, via HTTP and HTTPS, only to clients in the trusted range of IP addresses
- a missing systemtopo.txt file is not considered an error any more, and will not make the Web client fail
- you can now create an img/ sub-directory under the project's main directory. Files in this subdirectory have precedence over files in the root /img directory
- support for the Elsner P03/3 Modbus weather station
- updated ParadoxEVO driver

EVENTS:

- New events for temperature control central unit status:
 - TEMP <unit> = <WINTER | SUMMER>
 - TEMP <unit> = <OFF | PROT | MAN | STD 1 | STD 2 | STD 3 | HOLIDAY>
- MAIL action: you can optionally specify the destination SMTP server name or IP address by appending :<server name or address> to the recipient's email address, for example: john@example.com:192.168.1.1
- you can now check if a variable is defined: the event condition \$<varname> = "" is true if the variable is not defined or if has been explicitly set to ""
- new DATALOGGER action:
 - support for COUNTER and RANGE data modes, e.g.:
 - DATALOGGER <name> [<name>] = INIT COUNTER [<upper limit> [<decimals> [<hour interval>]]]
 - DATALOGGER <name> [<name>] = INIT RANGE [<decimals> [<hour interval>]]
 - support for CSV statistic and log files:
 - DATALOGGER <name> [<name>] = FILE LOG <filename> [TIMESTAMP] - appends most recent data to a log file, with optional timestamp
 - DATALOGGER <name> [<name>] = FILE STAT <filename> - creates a new file with all statistic data (same data as current chart)
 - DataLoggerCsvSeparator = <tab | comma | semicolon> parameter in hsyco.ini

Java API:

- new ping(String host, int timeout) utility method. Returns true if the host is reachable, false otherwise. host is the hostname or IP address; timeout is the timeout in milliseconds
- sendMail() method: you can optionally specify the destination SMTP server name or IP address by appending :<server name or address> to the recipient's email address, for example: john@example.com:192.168.1.1
- adds user.java callback method WebRootRequestEvent(InetAddress addr, boolean secure, String useragent). Called when the Web server receives a root URL request, can return a string to redirect the browser to a valid URL, or null to prevent redirection
- new static void user(String name, String param) command method. Triggers a USER event and the userCommand(String name, String param) Java method. Can be used as a calling mechanism between Java and EVENTS.
- new scheduler processor, allows to call user methods at configurable intervals. You define schedules using a group name and a schedule name. Schedules under the same group run in the same thread and are executed sequentially, based on their interval in milliseconds. Schedules in different groups run in parallel:
 - user.SchedulerEvent(String groupname, String schedulename) callback blocking method
 - user.schedulerRegister(String groupname, String schedulename, int interval) command method, to register a new scheduled callback; interval is defined in milliseconds
 - user.schedulerRemove(String groupname, String schedulename) method to delete a schedule
- MODBUS I/O methods readHoldingRegisters(), readInputRegisters() and writeMultipleRegisters() can now be invoked passing a name parameter that represents either a predefined I/O Server, or the host name or IP address of the MODBUS gateway. This makes possible to call these methods without defining an I/O server for each gateway
- new DATALOGGER API:
 - static boolean dataLoggerInit(String type, String name, double counterUpperLimit, int decimals, int hourInterval)
 - static boolean dataLoggerRefresh(String name)
 - static boolean dataLoggerUpdate(String name, double value)
 - static boolean dataLoggerClear(String name)
 - static boolean dataLoggerSave(String type, String[] names, String path, boolean timestamp)
 - DataLoggerCsvSeparator = <tab | comma | semicolon> parameter in hsyco.ini

index.hsm / GUI:

- access control: an administrator cannot change its own access rights
- support for AVI video files generation from recorded images
- when a camera frame is displayed, if the camera feed is standing-by and the last available frame is old, a blank frame is returned until a new frame is retrieved from the camera. It is also returned if the camera is offline
- (link), (linkmini), (linkmicro), (imagelink): you can specify an absolute or relative URL in the <page> parameter. The tag will work as an <A HREF> tag, loading the new page. If the loaded page is an HSYCO menu, the [< back] link will show up in the menu page, to reload the referrer page
- new (userimagelid file; pos; width; height; name; param; text) tag
- new (video src; pos; width; height [, mode]) tag for playback of HTML5 supported video formats. This tag has the identified version, and allows dynamic control of position, visibility, play/pause and mode
- (securitylog) has a new format and is now the same for all skins. The new format is (securitylog <id>; <pos>)
- (securitydisplay) now has a transparent background, rather than the blue, display-like background of previous versions. Draw a panel behind the (securitydisplay) object to replicate the original graphics
- (securitylink) and (securitypage) have been removed. Use a normal page instead, with (securitylog) and (securitydisplay), even on fixed size skins
- (templink), (tempdlink), (temppage), (musicpage), (timerlink), (timerdlink), (timerpage) objects were deprecated in version 2.9.0 and have been removed
- new object (sliderlid address; pos; label). Values set with uiSet(<id>, "value", <value>): 0% - 100%; 0.00 - 1.00; 0/<Max> - <Max>/<Max>, on, off (case insensitive)

Bug-fix:

- several stability and performance fixes in the camera frames acquisition processes
- frame acquisition now works also for cameras returning single jpeg frames without the content length header
- fixes to the systemtopo.txt parser
- a bug in the MODBUS driver could prevent reading/writing data for devices with MODBUS address greater than 127
- PROGRAMTIMER events were disabled when running HSYCO with the VIDEO license
- fixed PIN/PUK keypad problem with Android clients
- the HSYCO Server process automatically quits and restarts in case of "too many open files" exceptions

2.9.0

User Interface:

- changes and new features for displaying cameras, including a new drop-down selection list of motion events. It is also possible to freely display cameras in any page
- pop-up pages in the variable size skin are now positioned above or below the link button if there isn't enough space to the left or right
- improved response time of PIN/PUK keypads for devices using iPhone OS 3.x
- if the PIN appears because the browser is connected to a not trusted IP and the session is already authenticated but expired, if the IP changes to a trusted IP the PIN panel is canceled and the session resumed automatically
- if the authentication request fails after entering the PIN or PUK, you can press C to return to the PIN entry dialog at any time, or automatically after 30 seconds
- on loading a page from the cache, if the connection to the server cannot be established, a dialog box appears with a button to force a page reload
- during cameras playback, the "Auto" button is replaced by an Email button. Pressing this button pauses the playback and shows a pop-up where you can type an email address to send a mail with the frame attached in its native resolution. You can also right click or click and hold on the frame when paused to save the image locally
- obsolete skin objects: (musiclink), (musicpage)

Core:

- new format for the systemtopo.txt file and automatic generation of the devices and temp sections, based on auto-detection of existing devices; to enable the automatic discovery and generation of the systemtopo.txt file, set openServersDiscovery = true in hsyco.ini
- support for the Ekahau RTLS Position Engine and location tags, including buzzer and display control for the T301BD tag
- performance and stability improvements in cameras image processing
- in the trustedNet parameter in the hsyco.ini file, you can now enter multiple, comma-separated IP ranges or individual IP addresses
- add-on class Tecnoalarm (TECNO OUT driver) is now included in the HSYCO distributions
- add-on class Bentel (KYO320 driver) is now included in the HSYCO distributions
- the ParadoxEVO, Bentel and Tecnoalarm drivers are now fully integrated in HSYCO. If the configuration files are found at start-up, the classes are

automatically instantiated and there is no need to add any Java code in user.java to make them work

- when HSYCO stops automatically after one of the files defined in AutoKillFiles changes, all camera recording markers are saved and quickly restored from the database when HSYCO restarts. If it restarts after a forced quit, markers are rebuilt based on the actual frames stored in the motion directory
- when HSYCO starts, the SSL certificate is checked and, if the name set in the ServerName parameter in hsyco.ini doesn't match the certificate name, a new self-signed certificate is automatically generated to replace the old one
- support for "HSYCO Mini" license rights. A new "limits" attribute in license.txt is used to enable the Mini restrictions. At this time the supported keywords are 1CAM, 4CAM and 1DMX
- we had hard thresholds of 6GB and 3GB of free disk space; below these limits, HSYCO would start to drop older frames, or stop recording new frames. Now if the motion directory has its own dedicated partition, the thresholds are 2GB and 1GB, while if the partition is the same as the HSYCO root directory, they are set to 15% and 10% of total disk space for that partition

User Interface Objects:

- (camera!id <camera id>; <position>; <width>; <height>; <destination panel id>) live display, with click to camera page. The <destination panel id> is optional; if defined, clicking on the camera image changes the camera displayed on the destination camera panel; the <camera id> is either numeric or the camera name or grid name
- (camerapanel!id <camera id>; <position>; <width>; <height>; <camera list>) live display within panel and in-place ptz and playback controls. The <camera list> parameter is optional, and can be used to restrict the list of cameras to show in this panel; the camera id is either numeric or the camera name or grid name
- (cameraimage!id <camera id>; <position>; <width>; <height>) live display, no click, no controls; the camera id is either numeric or the camera name or grid name
- the <camera id> parameter in the (cameralink) object can be numeric or the literal camera name or grid name
- the (cameragridlink) object is obsolete and will be removed in HSYCO 3.0. Use (cameralink) instead
- the (cameralink) object has the optional parameter <destination panel id>: (cameralink <num_cam>; <position>; <color>; <text>; <destination panel id>). If defined, clicking on the button changes the camera displayed on the destination camera panel

- new (container <position>) and (endofcontainer) convenience objects. These objects are used to group other objects. The objects in a container have the position relative to the container position. Containers can be nested. With identified containers you can also control the visibility and position of the containers and all objects inside
- new (chart) object
- new optional CSS style attribute for the text object: (text <position>; <text>; <css>)
- individual parametri of objects in index.hsm can be nested with () parentheses to allow for semicolon inside a parameter
- new objects for forms support:
 - (input!id pos) and (input!id pos; css) input fields; uiSet() can be used to set the text and visibility of input fields
 - (submit!id pos; color; label) for the form submit button. When the submit button is pressed, all input fields and their ids are passed to userButton() or virtualRemote(); with userButton() you can control a page change on submit. (submitmini) and (submitmicro) objects are also available for small and very small buttons
 - new optional attribute color for the panel object: (panellid pos; width; height; color); supported color values: g,r,b,gr,y
 - new object (autombig), same syntax of (autom), shows a large button with areas for up/open, stop and down/close
 - new object (marqueelid <pos>; <width>; <height>; <direction>; <speed>; <text> [; css]). It is just like a text object, but with scrolling
 - the new directive (#deviceimage disable) can be used to disable the pop-up with the image or camera associated to device icons; works with all skins
 - the new directive (#scale f) allows you to define a fixed scaling factor for classicbig skins. f is a decimal number greater than 0
 - the new directive (#kiosk), only available in the classicbig skin, allows you to present a page without the HSYCO menu bar and borders. (#kiosk lock) also blocks users from changing page
 - (cameralist), (cameragridlist) and (cameraoverlay) are now meta directives: (#cameralist), (#cameragridlist) and (#cameraoverlay). The old format is supported for backward compatibility but is deprecated, and will be dropped in the next major release

Java Programming:

- new callback method SunPositionEvent(int azimuth, int elevation)
- new callback method userCommand(String rem, String id). This method will replace virtualRemote(), and allows you to return a page name, forcing the GUI to change page in response to a user button's click

- new callback method `programTimerEvent(String name)` called on program timers execution
- new methods `modbusReadHoldingRegisters(String name, int unit, int address, int quantity)`, `modbusReadInputRegisters(String name, int unit, int address, int quantity)` and `modbusWriteMultipleRegisters(String name, int unit, int address, byte[] bytes)` for reading and writing MODBUS registers on MODBUS/TCP I/O servers
- new methods `varGet(String name)` and `varSet(String name, String value)` to read/write program variables. Variables names ending with ! are considered persistent and their values are preserved when HSYCO restarts; normal variables are deleted when HSYCO starts
- new methods `programTimerSet(String name, int seconds)`, `programTimerClear(String name)`, `programTimerReset(String name, int seconds)` and `programTimerRepeat(String name, int seconds)` to manage program timers
- the `uiSet(id, attr, value)` now supports the ability to force a page change from user code. You have to set the menu name (the name of the directory containing the `index.hsm` file) as `id`. Setting `attr` to "page" and `value` to the page id changes to that page; setting to an empty string changes to the previous page. Setting `attr` to "lock" and `value` to "true" blocks the user's navigation; set `value` to "false" to unlock
- `uiSet()` can be used to set the camera id (the attribute name is `camera`, and should be set to the camera number or name), visibility and position of (`camera`), (`camerapanel`) and (`cameraimage`) objects
- `readComm()` and `writeComm()` now write the full trace of received and sent bytes to the log file only if both `verboseLog = true` and `userLog = true`
- a new method `public static int sendMail(String to, String from, String subject, Vector<String> body)` overloads the original `sendMail()` method, allowing you to send multi-part emails, with mixed text and camera images. To send an ordinary text, just fill the vector with the desired text string. To send an image, add a string to the vector with the following format: "cam:<cameraname>[:<seconds_back>]". For example, "cam:door" sends a live frame from the camera called "door"; "cam:door:2" sends a frame that was recorded two seconds before the last recorded frame; "cam:door:0" sends the last recorded frame. You can send multiple images in the same message, and mix text and images
- new methods `getBentelPlugIn(String id)`, `getParadoxEVOPlugIn(String id)` and `getTecnoalarmPlugIn(String id)` to retrieve the instance objects in `user.java` and use the specific API methods

Events Programming:

- new events interpreter, with support for complex events and scalar values comparison
- new events and action verbs
- improved performance

Bug-fix:

- improved reliability of the sendMail() method
- recording when the Rovio mobile camera moves doesn't stop any more while the camera is still moving
- fixed readComm() method to improve reliability when waiting for input data
- fixed a bug that could affect the reliability of the event monitor of OpenWebNet when using multiple servers; also improved the error messages of the OpenWebNet Monitor engine
- fixed a bug in LocationEvent() callback method that caused an incorrect long representation of the MAC parameter

2.8.4

Core:

- bidirectional serial port support for the RS232 port of IRTRans
- support of CCF hexadecimal strings for IRTRans
- new method public static int deviceFunctionGet(int device)
- new method public static int sendMail(String to, String from, String subject, String body)
- add-on classes Netstreams and ParadoxEVO are now included in the HSYCO distributions
- support for the HWg Damocles line of products, with input lines counters

Bug-fix:

- in some occasions, a bug in the Open Web Net monitor thread could cause issues when using multiple gateways
- the IO id = FLIP action in events.txt could generate a parse error and cause the event to be ignored at start-up
- the TOGGLE action of openCommand() didn't work properly
- powerSet(int power) now works even if no OpenWebNet server is defined in hsyco.ini
- using the classicbig skin, the camera page didn't display camera images after a rotation and subsequent switch between landscape and portrait page formats
- improved stability of PIN/PUK authentication
- because of a bug in the Syslog service parser, call events generated by the Grandstream GXE502x IPPBX didn't work

- in the TimeEvent(long time) user callback, the time parameter was not correctly aligned on the current time
- a bug in the WebServer's page parser caused parse errors when running the HSYCO Server on Microsoft Windows

2.8.3

GUI:

- (menu#portrait) (menu#landscape) (page#portrait) (page#landscape) .hsc files to support automatic page reformatting based on orientation --- for variable size skins only. On orientation change, look for the corresponding format, then the base format and finally the other orientation format
- (link!id), (linkmini!id), (linkmicro!id), (dlink!id), (image!id), (imagelink!id) support for dynamic attributes

Core:

- fully integrated support of the serial port in HWg I/O Controller
- improved reliability of files synchronization process in High Availability configurations
- the status loop in the GUI core has been optimized
- changes to the off-line cache manifest naming, for faster reloading of changes

2.8.2

API:

- new setPower() utility method
- the CameraCommandEvent() callback method is now called for all cameras with a PTZ driver defined in hsyco.ini, not only for PTZ=user. If the method returns -1, the PTZ driver command is skipped

events.txt:

- new CAMERACOMMAND event, called for all cameras with a PTZ driver defined in hsyco.ini. If the event matches, the PTZ driver command is skipped

GUI:

- new (lighticon) and (automicon) skin objects, with solid gray or glass semi-transparent effect
- new (background) object
- new status icons in camera pages for the Rovio camera

Bug-fix:

- because of a bug in the GUI code, cascaded pop-up windows didn't work properly. You could not use a (link) object inside a pop-up to open another pop-up
- multiple (text!id) or (user!id) objects with the same id were not handled correctly
- device timers (the off timers, usually set with `TIMER d = SET s, deviceOffTimerSet(int device, int seconds)` etc.) were not disabled in High Availability configurations when the server is not active. Now the execution of the off command is disabled if the HA state is not active
- minor GUI fixes:
 - in the classicbig skin, the micro pop-up that shows the image associated with a device were not handled correctly on Safari Mobile. Now they work properly, disappearing after a few seconds from the key press. They also have the X corner to close manually
 - Rovio status icons moved up a few bits in templates, classic and modern skins, to avoid overlapping with the command message
 - Rovio status icons temporarily showing on other cameras
 - pop-up GUI items not closing on session time-out events remained visible in the log-in window

2.8.1**Core:**

- performance optimization for I/O devices state machine and events propagation

Security:

- `KeysTrustedValidityHours` and `KeysNotTrustedValidityHours` parametri in `hsyco.ini` also accept time in hh:mm format
- new inactivity session expiration logic, configurable with `KeysInactivityHours` and `KeysInactivityMode` in `hsyco.ini`

events.txt:

- multiple identical events can be declared in `events.txt`

GUI:

- Web interface network settings functions can be disabled setting `WebAdminNetConfigLock=true` in `hsyco.ini`

2.8.0

Core:

- High Availability master-slave configuration support
- support for HTML 5 off-line cache storage, dramatically improving page loading performance when accessing HSYCO using slow connections. Works with iPhone 3, Safari 4 and Firefox 3.5. the OffLineCache parameter in hsyco.ini can be used to enable/disable this feature
- support for DMX512 universes (using the KISS-BOX Ethernet Gateway)
- userLog parameter in hsyco.ini to enable/disable logging of user.java and events.txt command calls
- support for HWg I/O controllers and monitoring interfaces

Network cameras:

- several changes in the cameras frame acquisition engine, to optimize performance, improve frame rate and reduce CPU load
- when available disk space drops below 6GB, approximately 20% of the oldest recorded frames for each camera are automatically deleted from disk. If available space drops below 3GB recording of new frames is temporarily disabled
- new Cameras.DroppedFrame.<camera id> parameter in hsyco.ini allows to set how many frames to drop for each camera during recording
- new CameraGrid.<id>.Resolution parameter to limit the maximum size of each grid
- new Camera.<id>.URL.Small parameter lets you specify a different URL to fetch frames at a lower resolution, in order to optimize acquisition performance
- support for PANASONIC PTZ cameras
- support for WowWee Rovio, including status, recording when moving, and update of home position
- semi-transparent standard command overlay on PTZ cameras, customizable with the (cameraoverlay) object
- pan, tilt and zoom movement is now continuous; the center quadrant of the image is used to stop the pan and tilt movement
- specific support for virtual PTZ cameras (Axis and Mobotix): top-left corner in the picture area zooms in about 60%; bottom-left corner zooms out to full-field (home) view
- camera rotation, both with the AUTO key and manual, is now constrained to the cameras contained in the grid. Direct access to a camera prevents rotation

API:

- new DMX512 control APIs
- new High Availability APIs

- readComm(String portName, int len) and writeComm(String portName, String data) for RS232 local communication ports support
- deviceOffTimerSet(), deviceOffTimerClear(), deviceOffTimerPreset() and deviceOffTimerReset() now also work for automation devices
- new cameraRecMode(String cameraName, boolean enabled) method to enable/disable camera recording
- new cameraMode(String cameraName, boolean enabled) method to turn a camera on/off
- cameraRecTrigger(String cameraName, String source, int seconds) now accepts seconds=0 to stop an active recording
- new cameraRecTriggerFull(String cameraName, String source, int seconds) method to record at full frame rate, ignoring the DroppedFrames option in hsyco.ini
- CameraCommandEvent() now supports move/stop event
- new sleep(long millis) method
- new temperature control system APIs: tempZoneGetTemp(), tempZoneGetSetpoint(), tempZoneGetOffset(), tempZoneGetMode()
- new uiSet() and uiGet() methods to dynamically change text, visibility, color and img attributes of (text!id), (user!id), (usermini!id) and (usermicro!id) objects
- new ioSet(), ioGet(), IOStartupEvent(), IOEvent() methods for I/O interfaces control
- new OpenEventFrame() callback method

events.txt:

- new UISET action to dynamically change text, visibility, color and img attributes of (text!id), (user!id), (usermini!id) and (usermicro!id) objects
- improved checks on zone number for TEMP commands (AUTO, STD, HOLIDAY)
- CAMERA name = sec action changed to CAMERAREC name = sec
- new CAMERARECFULL action to record at full frame rate
- the LIGHT d = FLIP, LIGHTGROUP d = FLIP, AUTOM d = FLIP = FLIP, AUTOMGROUP d = FLIP = FLIP actions now works also with multiple devices, with all devices following the state of the first device in the list
- new IOSTART and IO events, and IO action for I/O interfaces control

GUI:

- new fixed size skins: classic and modern
- new variable size skin: classicbig
- new skin objects:
 - linkmini linkmicro
 - userrgb usermini usermicro
 - irmini irmicro

- hbar, vbar
- image, imagelink
- text
- panel
- dimmer - controls SCS dimmer devices with a graphic object similar to (dmx)
- (cameraoverlay) - defines standard and custom overlays for PTZ cameras
- (text!id) (user!id) (usermini!id) (usermicro!id) - named objects with dynamic attributes that can be changed using UISET and UISet()
- popup (for classicbig only)
- musicsync (for classicbig only)
- securitydisplay (for classicbig only)
- obsolete skin objects:
 - templink, tempdlink, timerlink, timerdlink, temppage, endoftemppage, timerpage, endoftimerpage
- standard alarm log messages in the security system Web page can be replaced with customized messages defined in the (security) section of the systemtopo.txt file
- new administration pages to change OS's password and network settings
- improved handling of communication errors:
- if a communication error persists for about 30 seconds, the pop-up message has been removed and the page is automatically reloaded, but only immediately after loading; if a communication error occurs later, the send icon is raised and you can click on it to reload the web page, but there is no automatic reload
- new (#include <filename>) directive
- new (#size <page size>) and (#camerasize <width>x<height>) directives for classicbig
- multiple references to (light), (autom), (scene), (group) and (aux) objects are now possible without having to add the .x notation after the device id. The .x notation is still supported for backward compatibility

Bug-fix:

- fixed a bug that prevented access when using http://127.0.0.1/<pagename>/pin. from a loopback client (used to allow web clients running on the HSYCO server to use HTTP and omit the URLKEY)
- fixed a bug that prevented access when the URL had a query string parameter defined, like ?nopics or ?click
- if (security) was not defined in systemtopo.txt, the security log display failed on the Web page and errors could occur during state updates
- added support for serial devices with device names other than ttyS* on Linux (like USB-based serial ports)

- a bug in the OpenWebNet parser prevented the correct parsing of CEN frames on local SCS buses
- fixed a GUI bug during the PIN authentication phase, that could cause an unintentional login failure if roaming between untrusted and trusted client addresses occurred while the PIN page was displayed
- a bug in the index.hsm page parser caused a parse error for macro objects with empty last parameter, like: (page pagename;)
- a bug in the index.hsm page parser caused a parse error for files having a comment as last line
- the SCS startup procedure was changed to prevent a condition that, if the status of a device was changing state during the startup procedure, could cause a misalignment of that device's status
- SCS group commands were not handled properly by the monitor thread
- a bug in the events.txt parser prevented the correct match of IR events
- a bug in the events.txt parser prevented the correct match of events where the device id or channel had leading zeroes in the device number
- fixed a bug related to favicon.ico that could cause delays or hangups when accessing the HSYCO Web interface using some browsers

2.7.1

Core:

- location services: added support for Funkwerk GmbH access points

API:

- UserTimerEvent(String name, boolean active) is now also called once upon timer termination. The active parameter is set to true on activation calls and false on termination call. **WARNING:** this API change could affect user code compatibility running with previous releases of HSYCO core

GUI:

- the HSYCO web page is now automatically reloaded when any file changes in the www subdirectory
- on iPhone/iPod 3.0 or later the web page is not reloaded any more after a certain amount of camera frames have been loaded; the freeze issue has been solved on Safari 4.0 for iPhone
- improved handling of communication errors:
- if a communication error persists for about 30 seconds, a pop-up message asks to reload the page to try fix the issue (this is also needed when the page is loaded from the off-line cache, but the website certificate is not yet allowed - there is no other way to force the browser to ask permission to use a self-signed certificate)

- when the send icon is visible, you can click on it to reload the web page

Bug-fix:

- changed the text for manual zone temp setting in the Italian default skin
- multiple references to the same aux, group, scenario and cen object in index.hsm files now works properly, just like light and autom objects
- fixed a bug that could cause irregular or repeated calls to TimeEvent()
- openCommand (int function, int action, int device) now correctly supports AUX address commands
- improved support for high latency connections, including a 60 sec. idle timeout on HTTP server connections
- cameraCommandEvent() renamed CameraCommandEvent()
- OpenEventScene() and OpenEventAux() are now executed whenever a scene or aux frame is detected, not only on state changes
- OpenEventLight() and openEventAutom() are now executed only on state changes, not every time a frame is detected by the Open monitor thread

2.7.0

GUI:

- create, update and delete users
- define/change PIN and PUK
- grant administration rights to users
- selection of authorized Web pages for each user
- enable/disable users
- clicking on a dimmer button, when the status is on but the dimmer bar is not displayed, will bring the dimmer bar onscreen rather than turn off the light. This allows a smooth readjustment of the dimmer level, without having to turn the light off and back on again

Network cameras:

- redesigned navigation; server-based grid GUI (grid definition in hsyco.ini: CameraGrid.1 = c1 c2, c3 c4
- cameras GUI: PTZ control for Axis cameras and user customizable actions
- pre-recording buffer: the Camera.id.MotionBuffer = <frames> parameter in hsyco.ini can be used to set the number of frames to be recored in advance of any recording event. Default is to have the pre-recording buffer disabled
- image rotation: the Camera.id.Rotate = <degrees> parameter in hsyco.ini can be used to rotate camera images. Images are eventually cropped and filled with black side bars, depending on the original and final image size
- additional optimizations in image resize for variable, client-controlled frame size

- if images from a camera are not available when first polled from HSYCO startup, then a black empty frame is returned to the browser request, instead of a 404 Not Found error
- when a button with associated camera is pressed, the camera live thumbnail remains visible in the display area for 8 seconds, and can be clicked to go directly to the corresponding camera page
- licensing code to allow for distinct license keys for HSYCO and HSYCO VIDEO

API:

- void user.powerEvent() becomes int user.powerEvent() - if the method returns a not-negative integer, then this integer is set as current power state
- user.powerEvent() is now called for every power reading, even if the value is unchanged from previous reading
- new callback method user.OpenEventTempCentralUnit(int serverIndex, int mode, int value)
- new callback method boolean PBXCallEvent(String host, String caller, String called)
- new callback method public static int cameraCommandEvent(String function, String action, String camera) to process custom PTZ requests
- new method user.tempCentralUnitGetMode(int serverIndex)
- new method user.tempCentralUnitGetSettings(int serverIndex)
- new method openFrameCommand(int serverIndex, String frame) method to send generic frames to the SCS gateways
- new method public static int cameraCommand(String function, String action, String camera) to send PTZ commands to cameras
- new method public static int wakeOnLan(String broadcast, String address)

Security:

- access.ini flag to temporarily disable PINs (a # character at the beginning of the encrypted PIN)
- each individual user can now be restricted to a specific list of www directories when accessing the Web-based GUI

Core:

- full support of burglar alarm system, including a new security.log log file where all status changes and alarms are logged
- support for CEN commands
- ability to choose a specific template/skin in index.hsm: (#skin <skin directory>) -- default is templates; pic subdirectory moved under templates
- language directory in index.hsm (#language <language>) for the selection of text.txt file -- default is "it"; relocate text_<lang>.txt file in skin subdirectory
- relocate pin.html in skin subdirectory (also rename to pin.hsc)

- the embedded Web Server properly supports HTTP 1.1 HEAD requests
- implemented powerAdj “auto” value to automatically set power adjustment factor
- ability to define different port numbers for each OpenWebNet server, overriding the generic openServersPort setting in hsyco.ini
- the SqueezeCenter status monitor thread is now less verbose in the logs. When an error occurs, like when a device is off, no log is written unless verboseLog is set to true in hsyco.ini
- setting verboseLog=silent in hsyco.ini disables all log messages but the most serious errors and security messages
- when sending IR commands, there is an initial delay of 500ms before sending each command. Now it is possible to avoid this delay using the @0 directive in the command sequence

Bug-fix:

- fixed a bug that affected monitoring of GROUP AUTOM commands
- fixed a bug sending commands to devices with address A=0
- camera page header not properly refreshed when transitioning between first camera and first camera grid
- improved handling of nonexistent pages, now shows an alert message for debugging purposes, and also fixes a bug when reloading the GUI and the last current page has been removed
- hsyco.ini parametri values are now trimmed of leading and trailing spaces
- fixed a bug that could corrupt the current state information, causing the GUI to display a blank page, when parsing scenario configuration frames
- if powerAdj is not defined there should be no error message in the log file at startup
- OpenWebNet parser fix to avoid restart on lighting frames that break current frame format definition

2.6.1

Bug-fix

- when transitioning from a media page to the cameras page, the SCREEN: ON/OFF status flag could remain visible if a network error occurred while loading camera frames - the fix affects the JavaScript client code only

2.6.0

-
- Core:

- new software license key based on a unique hardware id. HSYCO Server will run with most monitor threads disabled and command execution disabled if license.txt is missing or invalid
- log files and folders, as well as camera recorded files and folders are now created with read/write permissions for all users, to simplify remote management from not-root users
- new HTML page template system and improved multi-language support
- httproot folder renamed www
- www files are now searched in the hsyco.jar package file if not found inside the www folder. All standard web client files, like templates, images, and JavaScript files are now distributed inside the hsyco.jar package
- added void user.StartupEvent() callback, called once at startup, before main threads are started
- added void OpenWebNetMonitorStartupEvent(int serverIndex) callback, called every time the Open Web Net monitor thread starts, after initialization, when monitoring is active
- added void user.TimeEvent(long time) callback, called every minute, at 00 seconds
- added measured temperature event callback function void OpenEventTemp(int zone, int temp), zone is the conventional zone number and temp is a positive number between 0 (0 C) and 500 (50.0 C)
- zones' set-point status temperatures are now updated much quicker than before
- the single thread used to fetch frames from the cameras has now been split into one distinct thread per camera, so that cameras' performance don't affect each other
- motion events markers implemented at core level
- HTTP Server performance improvements: content compression and HTTP/1.1 persistent connections fully implemented in the embedded HTTP server
- automatic Web page reload after HSYCO server restart
- user class now extends userBase skeleton class, and a number of stub methods and constants have been defined inside userBase to be used within user.java
- user call DaylightEvent() now not called anymore at HSYCO server startup
- CertCommonName parameter in hsyco.ini renamed ServerName; CertCommonName lookup performed for backward compatibility if ServerName not found

Access control:

- support multiple URLKEYs: the urlkey parameter in access.ini can be set as a list of comma separated URLKEYs; leading and trailing spaces for each key are ignored
- security fix: removed access key text and PIN from log files in case of access errors, leaving the first 6 characters of keys and the first and last digit of PINs
- by default, serving pages through the HTTP server is now disabled, except for localhost requests; use HTTPServerLowSecurityEnabled = true to fully enable the HTTP server, otherwise only requests for motion triggering will be processed (/x/camerarec)
- URLKEY is not required for localhost requests, a URL in the form /<path>/pin. is considered valid

GUI:

- dimmer level setting slider added for dimmer units
- overhauled << < > >> camera playback navigation buttons. Navigation is now event-based, forward and rewind buttons now allow for an accurate navigation through the individual events, while a sliding cursor shows the frame position within the event. The cursor can be clicked to jump frames within the same event with moving
- after a page reload, the GUI restores its state, to the sub-page and mode before reloading; GUI state is retained for 60 minutes
- workaround for iPhone 2.x Safari Mobile refusing to load images after a while. The client page is reloaded after a preset number of camera images have been loaded
- the keyboard can now be used to type digits when keypad is displayed (login page, timers and temperature control); allowed keys are 0-9, Backspace and Enter

Bug-fix:

- fixed PIN timeout page display clean-up and return to current page after successful login
- numeric keypad input now working properly; spacebar replaced C as alternative backspace key
- power meter also available in temperature control pages

2.5.4

Core:

- improved SCS monitor thread initialization - the monitor thread now requests device status three times, then continues instead of exiting. If the SCS server doesn't accept open access, but requires authentication, an appropriate error message is logged and the monitor thread quits

Access control:

- the authentication system based on two distinct pages, the pin.<urlkey> and index.html home page has been overhauled. Now the only client side url is pin.<urlkey>. When the client requests the pin page, if the key cookie is valid, the server directly serves the index.html content, skipping the redirection and subsequent loading of index.html. If the key cookie is not valid, the actual pin.html page is served as usual. Also, when the key expires, the client code directly jumps to the pin request panel, without reloading the pin.html page. Thanks to this, there is a significant time saving during the revalidation process after the key has expired

Bug-fix:

- a check on time validity has been added for timer set panels

2.5.3

Core:

- added support for multiple temperature control central units in multi-SCS environments
- added support for multiple power load control central units in multi-SCS environments
- electric loads control unit detection during Open Web Net monitor initialization, log of measured V, A, P
- server forced quit on critical files changes (autoKillFiles parameter in hsyco.ini defining the list of monitored files)
- improved location services - clients' location table is now checked and eventually updated during state update calls; MAC addresses have 10 hours expiration time
- improved update of temperature control's zones status after central unit commands
- self-generation of SSL certificates with appropriate common name based on server name if certificate not found
- changed default HTTP Server threads from 32 to 128
- support for zone parameter passed with camera triggers (camera=cameraName;zone=zoneName), also made available to user.CameraMotionEvent()
- automatic recording of motion events based on camera triggers can be disabled setting CamerasRecordingMotionTriggerSeconds = 0
- added public static void deviceOffTimerReset(int device, int seconds) method

Access Control:

- removed login/current trusted/untrusted network mismatch check and subsequent key removal

GUI:

- keys can be set as not-clickable using the NOCLICK option in systemtopo.txt
- keys can have a live camera associated in systemtopo.txt; hovering over the key or clicking it the associated live camera feed appears in the display area replacing the logo
- resized (smaller) camera images are used when the iPhone's orientation is vertical
- enhanced compatibility with iPhone 2.0:
- larger sensitive areas for logout, back, menu and location elements in the main display

Bug-fix:

- de-association location messages from access points should not overlap more recent location information. set location to unknown only if the de-association message comes from the same access point of current location
- camera frames time-stamp cookie should be set based on frame download time-stamp, not on current time
- changed OpenWebNet read timeout in hsyco.openDeviceStatus() to 10s
- user.CameraMotionEvent() now called once per trigger event, not every time a frame is saved
- reduced time needed to show location information
- improved responsiveness of camera playback << < > >> navigation buttons
- added remote IP Address in "POST method not supported" log message
- clicking a key with associated camera didn't work properly on iPhone, failing to execute command

2.5.2

Cameras:

- change to motion frames grabbing architecture: frames to be grabbed from normal live streams based on external HTTP trigger or API available to user callback code: cameraRecMode() to enable/disable camera feed recording; cameraRecTrigger() to trigger recording
- streams polling based on actual per-camera requests (live display or motion-triggered)
- remove camera images from motion DB after configurable expiration time (general default and camera-specific)
- display frame date/time during playback and current server date/time with live video

core:

- message and error messages logged to logs/yyyy/mmdd-message.log files instead of console

GUI:

- navigation history stack and standard back button on all pages
- AUX commands support (buttons with status and command execution)
- GUI container now centered inside browser's window
- favicon added

Bug-fix:

- GUI/camera page: when in PLAY mode, clicking on menu occasionally (and especially on slow connections) failed to make the last camera frame hidden, causing last frame to become visible on top of the menu page
- fixed bug in `SystemState.deviceOffTimerSet()` : calls to this method replaced the timer even when a timer was already present; now if a timer is already set, this call has no effect
- fixed navigation bug on music and temperature zone pages that prevented going back to the zones pages

2.5.1

Temperature control:

- support for central unit firmware v.3.0 (fix number of away days returned in monitor session)

Bug-fix:

- `hsyco.openDeviceStatus()` is now called for TEMP status initialization only if `tempZones` variable is initialized in `hsyco.ini` and server index is 0 for `OpenWebNetMonitor` thread
- GUI fix of -1 temperature display during status initialization

2.5.0

Access control:

- `keys.ini` file renamed `keys.data`

Core:

- timers/alarms management and GUI

Startup scripts:

- StartupItem script changed to log messages to console.log file in Hsyco home directory

2.4.3

Core:

- sun rise/set time (reset night sensor group at civil sun rise time, with adjustable offset)
- new GUI graphics

2.4.2

Location services:

- syslog server and parser for Apple Extreme and NETGEAR APs, with user event callback and configurable zone pages

Access control:

- automatic reload of access.ini file based on file's last modified time
- keys generated in a network range (trusted or not trusted) are only accepted within that same network range

Bug-fix:

- cameras playback GUI fixes

2.4.1

Core:

- Open Web Net auxiliary commands support
- process camera upload via HTTP, save images and manage system state information, with user actions callback
- motion video browser

Bug-fix:

- various fixes to improve compatibility with Symbian S60 browsers and when working in slow mode

2.4.0

Core:

- Multiple Open Web Net servers support

2.3.0

Access control:

- HTTPS server implementation

Bug-fix:

- SlimServer getState error when library not available

2.2.2

Music:

- new music interface, with zones selection and synch page
- added shuffle and repeat status
- new home theater interface
- new user callback for slim status change events

Bug-fix:

- synching a player with itself was allowed and created a problem with SlimServer. Now a self-synch command from the client is ignored and not sent to SlimServer
- text selection on double-click is now disabled
- IE7.0 links to menu and zones now work properly
- fixed tap highlight with iPhone

2.2.1

Security:

- extended key cookie hash
- 10 sec. delay on PIN/PUK authentication, synchronous block
- security logging of PIN, PIN/PUK logins, login failures and not matching key cookies

2.2.0

Security:

- Remote access support and user authentication

2.1.7

GUI:

- clicking on camera picture shows next camera
- fixed add button bug in music pages
- temperature controller fully implemented (central unit control and zones manual setting)

Bug-fix:

- fixed temp zones monitor of automatic/manual state

2.1.6

Core:

- Open Web Net monitor thread restarts if unable to load status at start-up
- fixed Emergency page link

GUI:

- state update timeout reduced from 15 to 12 seconds
- iPhone on-rotation scrolling below display when landscape
- fixed bug with display's button text clean-up timer
- temperature: central unit operation mode monitor

Music:

- Slimserver players controller implementation, with sync/unsync
- user events on power and volume change
- players status monitor logic completely redesigned

IRTrans:

- fixed closed connection bug when sending multiple commands

2.1.5

- fixed width issue on iPhone
- added dimmer status support, and on/off command
- added IRTrans support (with TCP monitor and UDP command connections)
- added Squeezebox display and control interface
- detect iPhone rotation event and adjust viewport for proper page width
- pages navigation
- stateDisplay(), looping through JSONState items rather than JSONTopo

- it is now possible to have a GUI with multiple buttons for the same OpenWebNet device

2.0

Cameras:

- camera integration with proxy features and image processing (2.0.3b)

GUI:

- added click, clickall, nopics flags
- full compatibility with IE7, Firefox, Safari, PS3, Nokia N80, iPod touch
- fixed bug with off timer, causing repeated off commands sent in some cases

HSYCO e Home Systems Consulting sono marchi registrati di Home Systems Consulting SpA. Mac, iPod touch, iPhone sono marchi registrati di Apple Inc. Java e JavaScript sono marchi registrati di Sun Microsystems Inc. Linux è un marchio registrato di Linux Mark Institute. Windows è un marchio registrato di Microsoft Corporation. IRTrans è un marchio registrato di IRTrans GmbH. SlimServer, SqueezeCenter, Squeezebox, Transporter sono marchi registrati di Logitech Inc. NuVo è un marchio registrato di NuVo Technologies LLC. AXIS è un marchio registrato di AXIS Communications. KNX è un marchio registrato di Konnex Association. DALI è un marchio registrato di ZVEI - Zentralverband Elektrotechnik- und Elektronikindustrie e.V. DOMINO e CONTATTO sono marchi registrati di Duemmegi srl. My Home è un marchio registrato di BTicino Spa. Panasonic è un marchio registrato di Matsushita Electric Industrial Co.Ltd. Mobotix è un marchio registrato di Mobotix AG. Rovio è un marchio registrato di WowWee Group Limited. HWg è un marchio registrato di HW group s.r.o. Kissbox è un marchio registrato di Kiss-Box B.V. Tecnoalarm è un marchio registrato di Tecnoalarm Srl. Paradox è un marchio registrato di Paradox. Bentel è un marchio registrato di Bentel Security Srl. Ekahau è un marchio registrato di Ekahau Inc. Altri prodotti e ragioni sociali possono essere marchi o marchi registrati di altre aziende e sono usati solo per esigenze dimostrative, senza intenti di violazione.

Home Systems Consulting SpA

www.homesystemsconsulting.com

