

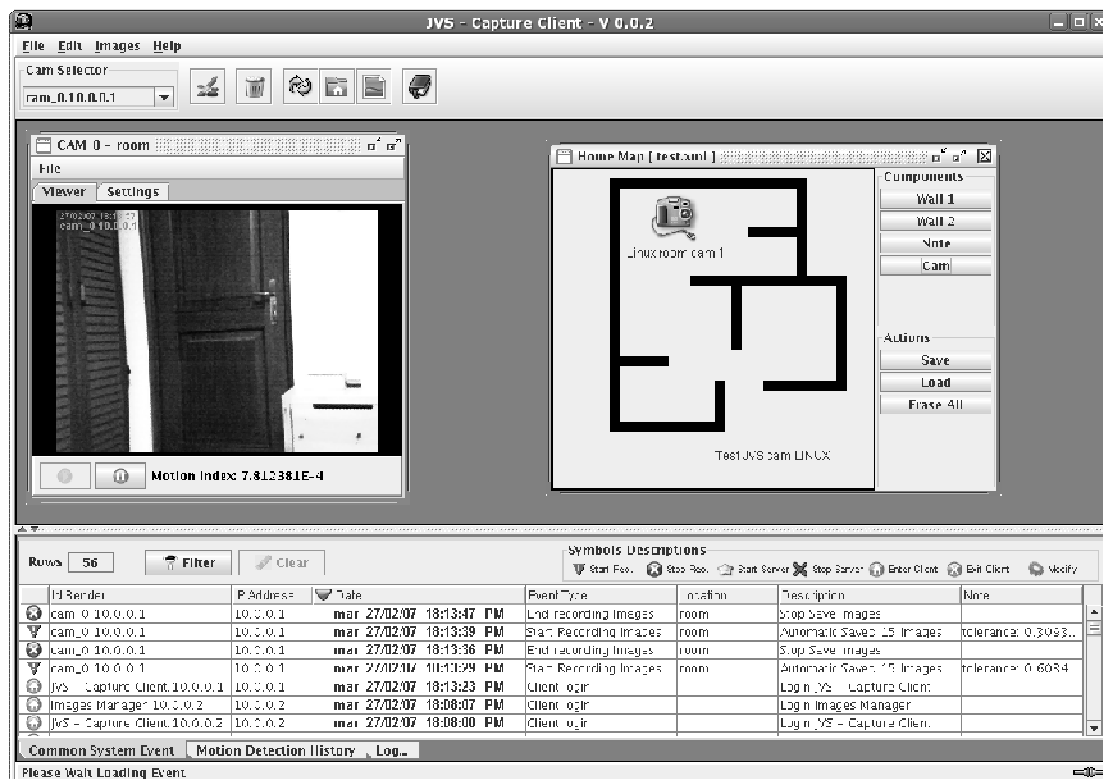
J.V.S. - Java Video Surveillance

<http://jvs.altervista.org/>

Manuale d'uso.

Introduzione

Il JVS e' un piccolo sistema di video sorveglianza multiplatforma scritto interamente in java(1) e sotto licenza GPL (<http://www.gnu.org/copyleft/gpl.html>). Il sistema si propone come una buona soluzione open source, per tutte quelle persone che necessitano di un piccolo e leggero programma di video sorveglianza. Il sistema JVS e' sufficientemente modulare e pensato per funzionare sia su una sola macchina che in una rete (sistema client-server). Viene distribuito per funzionare su più computer contemporaneamente; infatti e' possibile installare il server del jvs su una macchina, il grabber per la cattura delle immagini su un'altra ed eseguire l'applicativo client su un'altro computer ancora. Il software sfrutta le potenzialità del database HSQLDB, la libreria JMF(1) per la cattura e gestione delle immagini e JAI(1) per la manipolazione dei frames. E' possibile usare fino ad un massimo, non dichiarato, di tre webcam per computer, ed un solo server con a bordo il database. L'applicativo client offre la possibilità di modificare i parametri principali delle webcam e presenta un logger degli eventi di registrazione di immagini e di sistema, oltre ad un grafico delle registrazioni avvenute. Sempre nel client e' presente anche un visualizzatore di immagini remote, per vedere e salvare i frames delle webcams. Il server memorizza le immagini da registrare, cioè quelle dove e' stato avvertito movimento, e dà la possibilità di comprimere le immagini in un formato zip. Il sistema JVS non si propone come la soluzione ideale o "Open Source" per applicativi di video-sorveglianza, ma cerca di offrire una implementazione di base utile a tutte quelle persone che non possono permettersi costosi e migliori applicativi.



Installazione

L'installazione del JVS e' molto semplice, basta procurarsi una **Java Virtual Machine** (JVM) 1.5 o superiore e digitare da prompt dei comandi:

```
java -jar jvs-v.<versione>-install.jar
```

Apparira' l'interfaccia di setup, dove si richiede di inserire i parametri principali di configurazione.

Il JVS prevede l'utilizzo di librerie esterne per il corretto funzionamento. Tali dipendenze possono variare da versione a versione, pertanto e' consigliabile leggere il file **DEPENDENCIES.txt** presente sul sito, per sapere quali applicativi esterni installare. Tali dipendenze non sono fornite nell'installazione ufficiale di JVS.

Primo avvio

Dopo aver installato e configurato l'applicativo apriamo la cartella "bin" del JVS. Qui sono presenti tutti gli script di lancio. Esistono fondamentalmente due tipi di script: i .sh per sistemi Linux e i .bat per Windows. Ecco l'elenco dei file che si possono eseguire:

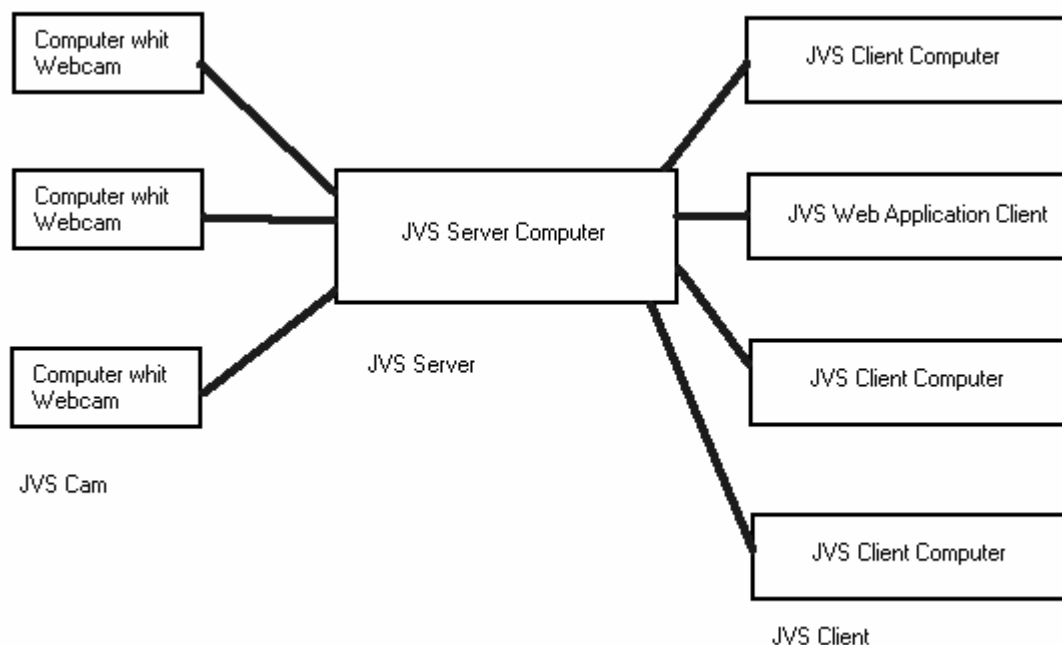
- jvs_cam.bat
- jvs_client.bat
- jvs_server.bat
- jvs_cam.sh
- jvs_client.sh
- jvs_server.sh

Per l'avvio sotto sistema Linux, assicurarsi che gli script .sh abbiano i permessi di esecuzione e digitare da schell il comando:

```
./jvs_server.sh
```

per attivare il server JVS. Stesso procedimento per tutti gli altri processi.

Il JVS prevede l'utilizzo di uno e un solo server avviabile con in comando "jvs_server", molte webcam, ognuna interrogata dallo script "jvs_cam" e vari client eseguiti con "jvs_client". Tutti questi processi possono essere eseguiti anche su diversi computer, per esempio una configurazione potrebbe essere:



JVS Server

Per eseguire il processo server basta lanciare lo script "jvs_server" relativo al proprio sistema operativo. Il server, prima cercherà di caricare tutti i dati salvati precedentemente nelle tabelle del database. Se non dovesse essere presente il db si cercherà di crearlo nella cartella "db" della vostra installazione.

Verrà aperta per default la porta **4516** per la comunicazione con i client, sia essi webcam o client jvs, e si aprirà anche la porta per la chat **4517**.

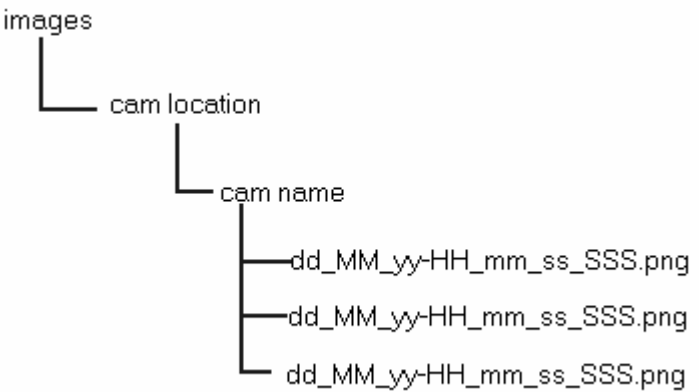
```
[28/02/07 12:37:22]:  
[28/02/07 12:37:22]:-----  
[28/02/07 12:37:22]:-      MAIN JVS SERVER V 0.0.2  
[28/02/07 12:37:22]:-  
[28/02/07 12:37:22]:- author : Cloud1979  
[28/02/07 12:37:22]:- email  : ricky9792002@yahoo.it  
[28/02/07 12:37:22]:-  
[28/02/07 12:37:22]:-----  
[28/02/07 12:37:22]:Server Name: SERVER_SOCKET  
[28/02/07 12:37:22]:Server Port: 4516  
[28/02/07 12:37:22]:  
[28/02/07 12:37:22]:-----  
[28/02/07 12:37:22]:
```

Qui invece si vede che il server ha ricaricato le tabelle già esistenti e avviato i 15 thread disponibili per i client.

```
[28/02/07 12:37:23]:Warning: Table Name: 'EVENT_TABLE' LOG: java.sql.SQLException: Table already exists: EVENT_TABLE in statement [CREATE CACHE  
TABLE EVENT_TABLE]  
[28/02/07 12:37:23]:Reload: Table Name: 'EVENT_TABLE'  
[28/02/07 12:37:23]:Warning: Table Name: 'CLIENTS_TABLE' LOG: java.sql.SQLException: Table already exists: CLIENTS_TABLE in statement [CREATE CACHE  
TABLE CLIENTS_TABLE]  
[28/02/07 12:37:23]:Reload: Table Name: 'CLIENTS_TABLE'  
[28/02/07 12:37:23]:Warning: Table Name: 'CURRENT_IMAGE_TABLE' LOG: java.sql.SQLException: Table already exists: CURRENT_IMAGE_TABLE in statement [CREATE CACHE  
TABLE CURRENT_IMAGE_TABLE]  
[28/02/07 12:37:23]:Reload: Table Name: 'CURRENT_IMAGE_TABLE'  
[28/02/07 12:37:23]:Warning: Table Name: 'MOTION_DETECTED_TABLE' LOG: java.sql.SQLException: Table already exists: MOTION_DETECTED_TABLE in statement [CREATE CA  
CHED TABLE MOTION_DETECTED_TABLE]  
[28/02/07 12:37:23]:Reload: Table Name: 'MOTION_DETECTED_TABLE'  
[28/02/07 12:37:23]:Start Server at port: 4516  
[28/02/07 12:37:23]: - Run client thread num: [ 0 ]  
[28/02/07 12:37:23]: - Run client thread num: [ 1 ]  
[28/02/07 12:37:23]: - Run client thread num: [ 2 ]  
[28/02/07 12:37:23]: - Run client thread num: [ 3 ]  
[28/02/07 12:37:23]: - Run client thread num: [ 4 ]  
[28/02/07 12:37:23]: - Run client thread num: [ 6 ]  
[28/02/07 12:37:23]: - Run client thread num: [ 7 ]  
[28/02/07 12:37:23]: - Run client thread num: [ 8 ]  
[28/02/07 12:37:23]: - Run client thread num: [ 10 ]  
[28/02/07 12:37:23]: - Run client thread num: [ 11 ]  
[28/02/07 12:37:23]: - Run client thread num: [ 9 ]  
[28/02/07 12:37:23]: - Run client thread num: [ 14 ]  
[28/02/07 12:37:23]: - Run client thread num: [ 13 ]  
[28/02/07 12:37:23]: - Run client thread num: [ 12 ]  
[28/02/07 12:37:23]: - Run client thread num: [ 5 ]  
[28/02/07 12:37:24]:The Socket Server is up.  
[28/02/07 12:37:24]:Chat - Current chat port is [ 4517 ]  
[28/02/07 12:37:24]:Chat - property is stored.  
[28/02/07 12:37:24]:EVENT_TABLE time to store: [ 16 ms ]  
[28/02/07 12:37:24]:SERVER READY
```

Da notare il messaggio SERVER READY e il tempo in millisecondi che si impiega ad effettuare un salvataggio di un evento nel database. E' possibile usare l'opzione "servername:porta" dove servername e' il nome del vostro server JVS.

Tutte le immagini registrate dalle webcam vengono salvate nelle cartella “images” del db creata a runtime. All’interno di tale cartella vi saranno tante directory con lo stesso nome del valore della locazione impostata per il processo jvs_cam e internamente tante directory quante sono le webcam inserite in tale locazione.



Come si puo’ notare tutti i nomi delle immagini sono salvate con la data della loro cattura, nella forma:

dd	Giorno
MM	Mese
yy	Anno
HH	Ora
mm	Minuti
ss	Secondi
SSS	Millisecondi

Tutte le immagini sono in formato png.

JVS Cam

Il processo jvs_cam e’ usato per connettersi alla webcam. Nello script “jvs_cam” sono elencate le variabili che potete configurare sia in fase di installazione sia con un qualsiasi editor di testo. Ecco un esempio:

```
cd ..

set JAVA_EXE=${JAVA_HOME}\bin\java
set BOOT_CLASS=org.jvs.client.grab.ImagesGrabber

set IP_ADDRESS=${cam.host}
set PORT=${cam.port}
set CAM_NAME=${cam.name}
set CAM_LOCATION=${cam.location}
set MOT_DET=${cam.mot_det}
set RUSH=${cam.rush}
set NOISE_R=${cam.noise}

set PARAM=%IP_ADDRESS% %PORT% %CAM_NAME% %CAM_LOCATION% %MOT_DET% %RUSH% %NOISE_R%

set BOOT_CLASS=org.jvs.client.grab.ImagesGrabber

set JVS_LIB_PATH="./lib/jvs.jar;./lib/jai_codec.jar;./lib/jai_core.jar;./lib/jmf.jar;./li
%JAVA_EXE% -xms64m -xmx256m -classpath %JVS_LIB_PATH% %BOOT_CLASS% %PARAM%
```

Le variabili \${JAVA_HOME}, \${cam.host}, ecc... sono configurate dal programma di installazione, ma nulla vieta di creare tanti script di lancio per le vostre webcam con valore di parametri differenti. L’importante e’ che il valore della variabile CAM_NAME sia sempre univoco.

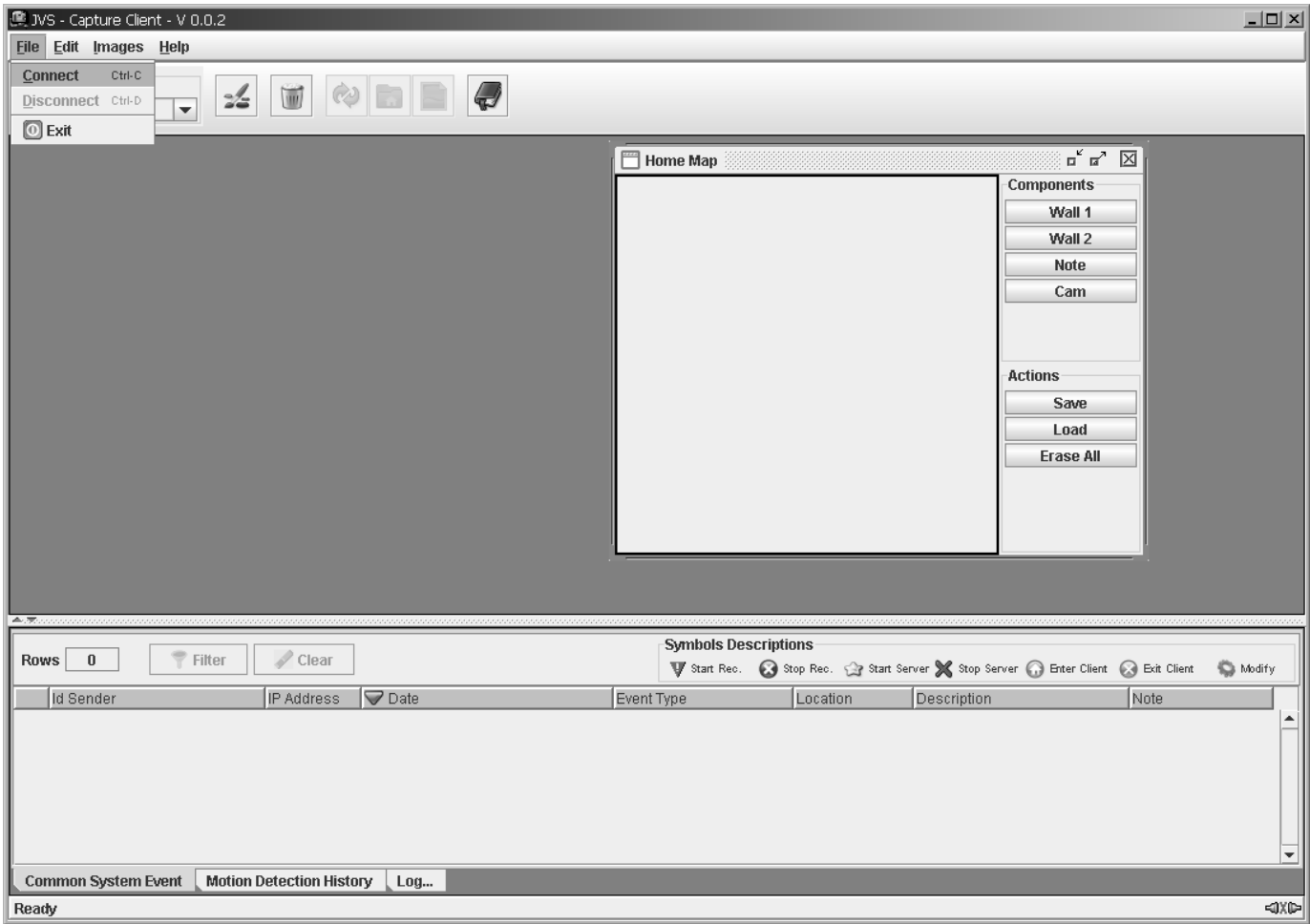
Ecco in dettaglio la lista delle variabili:

JAVA_HOME	Directory della java virtual machine
IP_ADDRESS	Indirizzo IP del server JVS
PORT	Porta del server JVS
CAM_NAME	Nome univoco della webcam
CAM_LOCATION	Locazione della webcam
MOT_DET	“on” se si deve attivare il sensore di movimento, “off” altrimenti
RUSH	Tempo in millisec. Di registrazione frame quando c’e’ movimento
NOISE_R	“on” se si vuole il soppressore di rumore delle immagini, “off” altrimenti

In questo script come negli altri vi e’ l’elenco delle librerie esterne che vengono usate dal sistema. Tali librerie non vengono fornite dal pacchetto di installazione del jvs, ma debbono essere scaricate e installate manualmente.

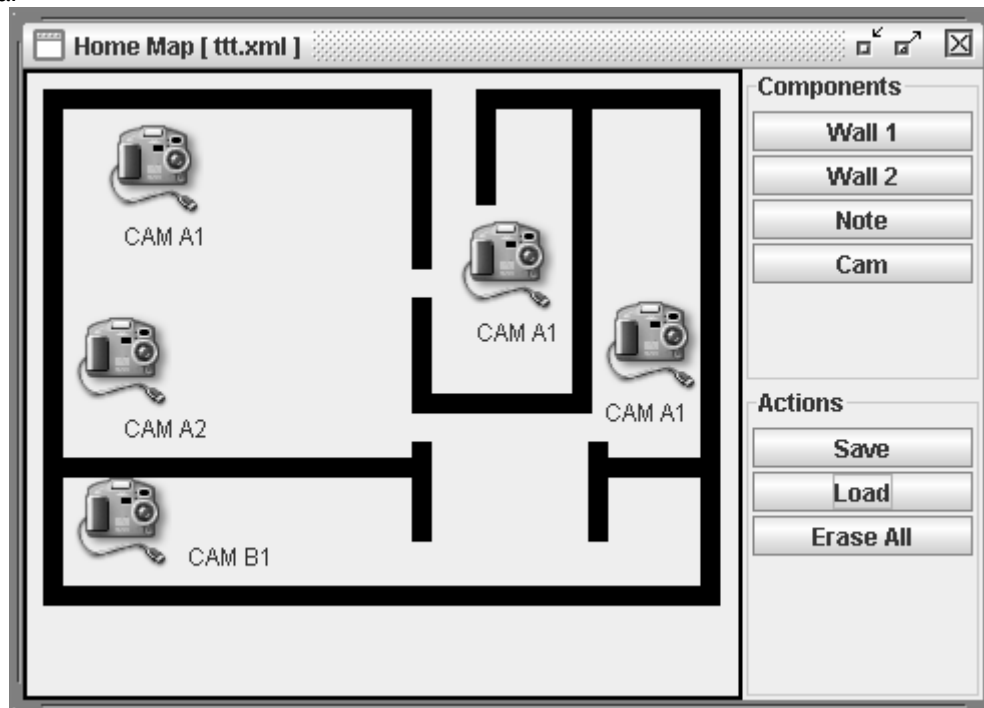
JVS Client

Per attivare il client JVS basta lanciare lo script “jvs_client”. Apparira’ l’applicativo client senza essersi connesso al server, per connettersi e ricevere le immagini delle webcam premere dal menu File la voce Connect.

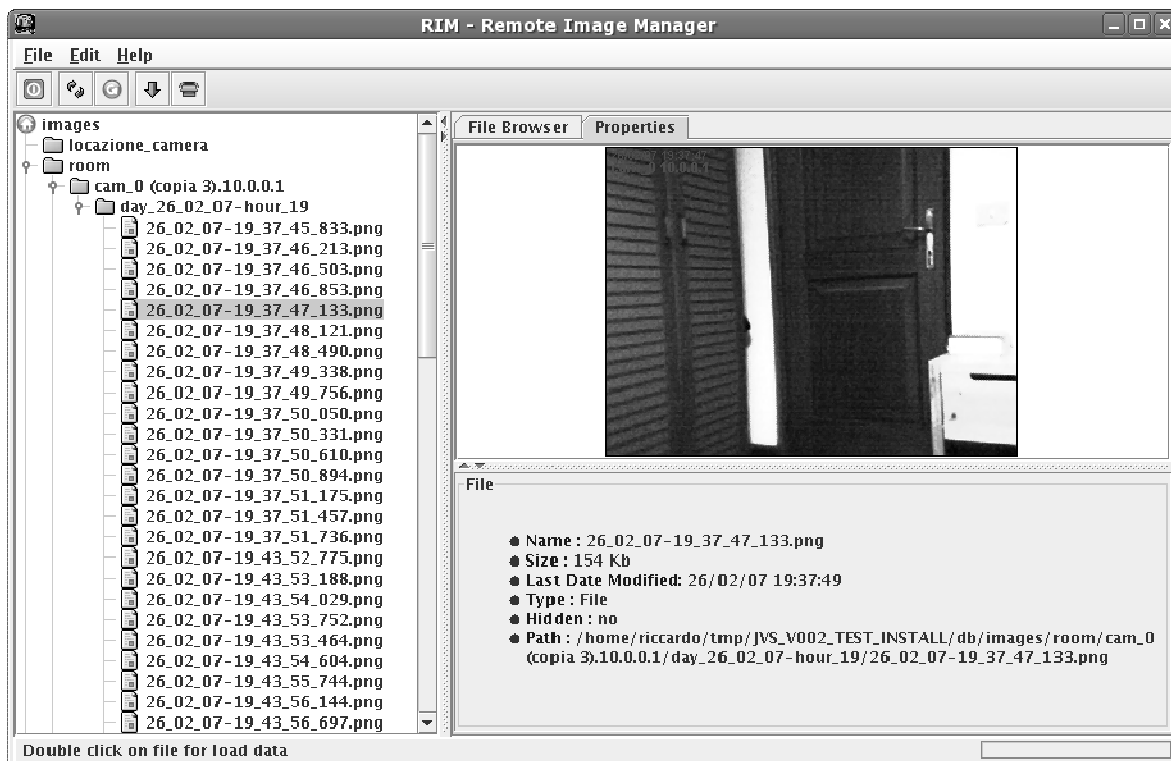


Inserito l'IP del server JVS e la porta, il client riceverà i dati e le immagini salvate nel server. Tutte le webcam sono configurabili anche a runtime dalla application client.

Sempre nel client è presente l'Home Map Editor, dove è possibile disegnare la propria piantina e collocare le webcam sulla mappa. Quando una webcam segnala un movimento, la cam corrispondente sulla mappa verrà cerchiata di rosso per indicare che c'è stata una anomalia in quella zona.



Oltre ad un piccolo viewer per le immagini catturate è presente anche l'applicativo RIM (Remote Image Manager) per la cancellazione, creazione di file compressi zip e scaricamento di dati dal server al client.



Nella directory di installazione sotto la cartella “web” vi e’ il file **javswc.war**. Questa e’ una applicazione client web utilizzabile con un web sever che permetta l’utilizzo delle jsp.

Applicazioni Risorse Desktop

mer 14 feb, 19.53

J.V.S. Client Web V0.0.1 - Mozilla Firefox

File Modifica Visualizza Vai Segnalibri Strumenti 2

http://127.0.0.1:8080/javswc/jsp/InitPane.jsp?IP_SERVER=10.0.0.1&PORT_SERVER=4516

SUSE LINUX Entertainment News Internet Search Reference Maps and Directions Shopping People and Compan...

Configuration Properties

IP Server: 10.0.0.1

Port Server: 4516

Server Status: The server is up

Auto Refresh: 10 sec.


For contacts: ricky9792002[at]yahoo[dot]it

Actions

Drop DB Show Version Refresh Cams Logout


Messages

linux_cam



- IP Address:10.0.0.1
- Location:room
- Delay:500
- Motion Detection:yes
- Video Format:rgb - [320 x 240]
- Tolerance:0.195

secondcam



- IP Address:10.0.0.1
- Location:room2
- Delay:500
- Motion Detection:yes
- Video Format:rgb - [320 x 240]
- Tolerance:0.195

Rows: 92

Symbols Descriptions

Start Rec. Stop Rec. Start Server Stop Server Enter Client Exit Client Modify

-	Id Sender	IP Address	Date	Event Type	Location	Description	Note
/	secondcam.10.0.0.1	10.0.0.1	mer 14/02/07 19:53:18 PM	Start Recording Images	room2	Automatic Saved 15 Images	tolerance: 0.3323177
/	linux_cam.10.0.0.1	10.0.0.1	mer 14/02/07 19:53:17 PM	Start Recording Images	room	Automatic Saved 15 Images	tolerance: 0.36438805
/	secondcam.10.0.0.1	10.0.0.1	mer 14/02/07 19:52:38 PM	Client login	room2	Cam Capture Login: secondcam	-
/	linux_cam.10.0.0.1	10.0.0.1	mer 14/02/07 19:51:26 PM	Client logout	room	-	-
/	linux_cam.10.0.0.1	10.0.0.1	mer 14/02/07 19:51:23 PM	Client logout	room	-	-

Completato

jav-v.0.0.1... linux [linux] linux [linux] [JVS - Cap... [JVS_IMAG... webapps -... J.V.S. Clie... linux

(1)Java, JMF e JAI sono marchi/applicativi di proprietà della Sun Microsystems
(1)Java, JMF and JAI are Sun Microsystems trademarks.