**MANUAL DE INTEGRACIÓN DE IMÁGENES**

**CLIMAX – MASTERMIND**

**El presente documento contiene claves y su uso es confidencial destinado solo para uso interno.**

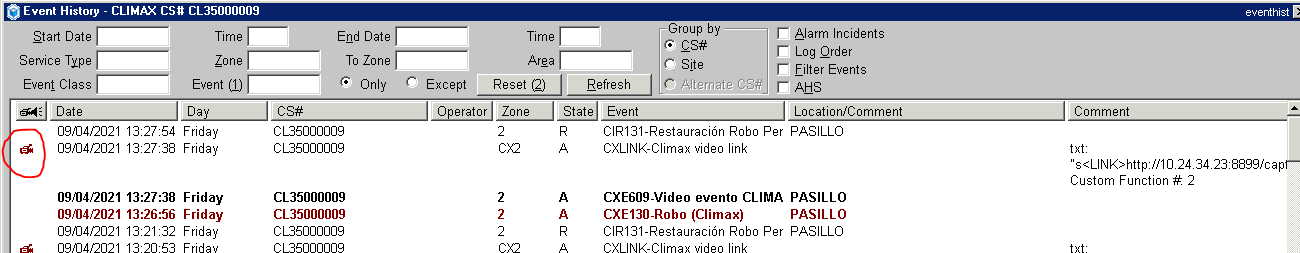
El presente manual tiene como objetivo ser una guía para la instalación y configuración de la integración entre la receptora Climax, y el sistema de monitoreo Mastermind.

Cuando un evento determinado que contiene la URL del directorio donde se alojan las fotos tomadas por el pircam de Climax llega a MasterMind, existe un stored procedure que se encarga de formatear esa url introduciéndola en el campo correspondiente AUX2 de la tabla event\_history del evento correspondiente, de manera tal que cuando el usuario observa el event\_history del cliente, verá un ícono con una cámara, que será un hipervínculo a la receptora que tendrá corriendo sobre un servidor Apache con PHP instalado, que contendrá el frontend para visualizar las fotos de manera práctica por el usuario.

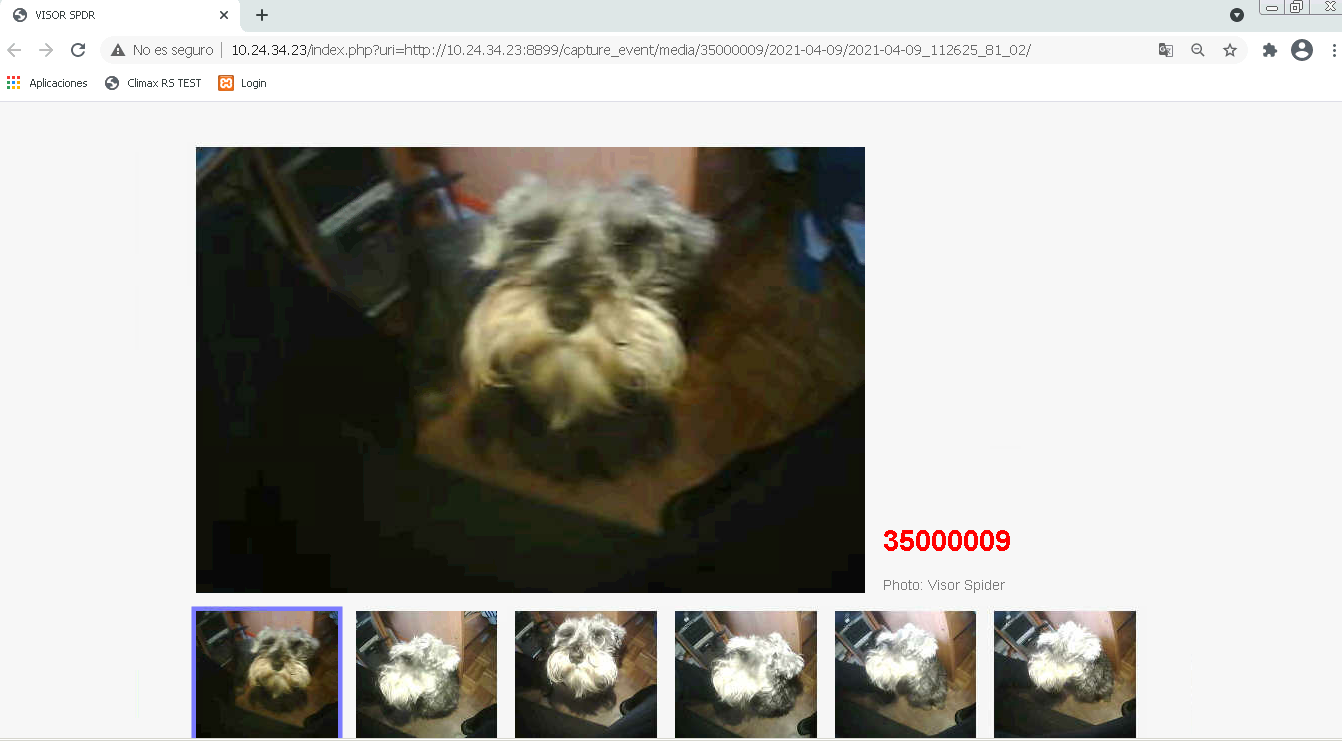
Ejemplo:

En este caso se trata del cliente cs\_no = CL35000009 montado en la base de test de España 10.24.33.140 monitordb.

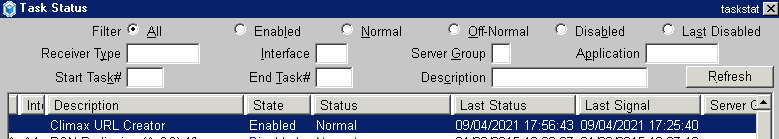
El evento configurado en esta base para recibir las imágenes es el CXLINK.



Clickeando sobre ese vínculo se abrirá el navegador que esté predeterminado, se recomienda Chrome, mostrando las imágenes asociadas al disparo de alarma.



El estado del servicio se monitorea por medio de una task de MasterMind, en la que se vé su estado, ultimo estado reportado y ultima señal procesada.

.

Componentes de la solución se encuentran en GITHUB en un repositorio privado al que se puede acceder con las credenciales prosegurteam/ pass = mastermind2021

Los componentes son:

1. Stored Procedure : ap\_climax.sql
2. index.php
3. Job.sql
4. Normalize.min.css
5. Style.css
6. Tarea en MasterMind

**1\_ Stored Procedure ap\_climax.sql**

Se ha descubierto que creando una url en el campo AUX2 de event\_history con un prefijo 'VF|MAS|', se genera un hipervínculo en el mismo, a esta conclusión se llegó observando el comportamiento de MASVIDEO.

El ap\_climax.sql ,SP en adelante, recibe como parámetro un numero entero, que representa la task\_no, este parámetro se pasará automáticamente por la tarea configurada en la ventana task status, al igual que lo hacen otros SP como ap\_o de las task 101 signal processor.

En esta primer versión 1.0 los settings correspondientes se configuran en el apartado settings del SP, (en la próxima versión se podrá hacer desde el task option).

|  |
| --- |
|  |
|  | Respetar formato hh:mm:ss.mmm\*/ |
|  | set @delayTime='00:00:10:000' TIEMPO QUE DESCANSARÁ ENTRE CADA ITERACIÓN |
|  | /\*minutos hacia atrás que mirará eventos a tratar a partir de inicio de programa\*/ |
|  | set @minutosHaciaAtrasInicial=1440 -- un dia por default al arranque, LA CONSULTA DE VERIFICACIÓN LLEGARÁ HACIA ATRÁS EN EL TIEMPO 1440 MINUTOS |
|  | /\*inicializar con las tasks por las que entra Climax\*/ |
|  | insert into @tasks\_no values (29), (229), (284) – LAS TAREAS POR LAS CUALES SE ESPERA QUE INGRESEN EVENTOS DE CLIMAX DE IMAGENES |
|  | /\*inicializar los eventos que traen imágen\*/ |
|  | insert into @image\_events values ('CXLINK'),('CXE609'),('CXE130') LOS EVENTOS QUE CONTENDRÁN URLS |
|  | /\*Etiquetas de inicio y fin del URL del directorio http\*/ |
|  | set @caracteriniciourl='<LINK>' |
|  | set @caracterfinurl='<LINK/>' |
|  | /\*El caracter PIPE\*/ |
|  | set @pipe='|' |
|  | set @barra='//' |
|  | set @prefijomas='VF|MAS|' |
|  | set @urlPhp='http://10.24.34.23/index.php?uri=' – HTTP:+IPSERVIDOR/index.php?uri= El servidor que aloja Visor Spider |
|  | set @maxRowsProcessing=200 – máxima cantidad de filas a procesar por cada iteracion |
|  | set @thisTask=isnull(@task\_no,1000); |
|  |  |

El algoritmo de la función consulta cada x (configurable) tiempo la tabla m\_signal\_processed, haciendo un join con event\_history sobre los eventos de video definidos que entraron por la tarea definida, todo usando nolock, en un periodo configurable, se ha probado con 24 hs y la performance es muy buena, no se percibe lentitud en MasterMind.

Entonces al detectar que un campo aux2 no ha sido formateado, lo formatea, habilitando así la funcionalidad de hipervínculo, que le enviará el hipervínculo a mostras al *Visor Spider,* representado por index.php para que muestre la información al usuario.

Al finalizar cada ciclo, coloca la tarea de MasterMind en “Normal”, si no pudiese hacer esa tarea durante al menos 1 minuto, MasterMind reportará “Process Error”.

**2\_ index.php**

Es el Visor Spider, está alojado en el directorio /var/www/html/ de la receptora, junto con sus hojas de estilo que le dan el formato de visualización (**Normalize.min.css y Style.css**).

Visor Spider, VS en adelante, es un programa realizado en PHP que recibe una url por método get en la variable **URI**, ingresa al directorio lista las imágenes y las presenta al usuario en su navegador web.

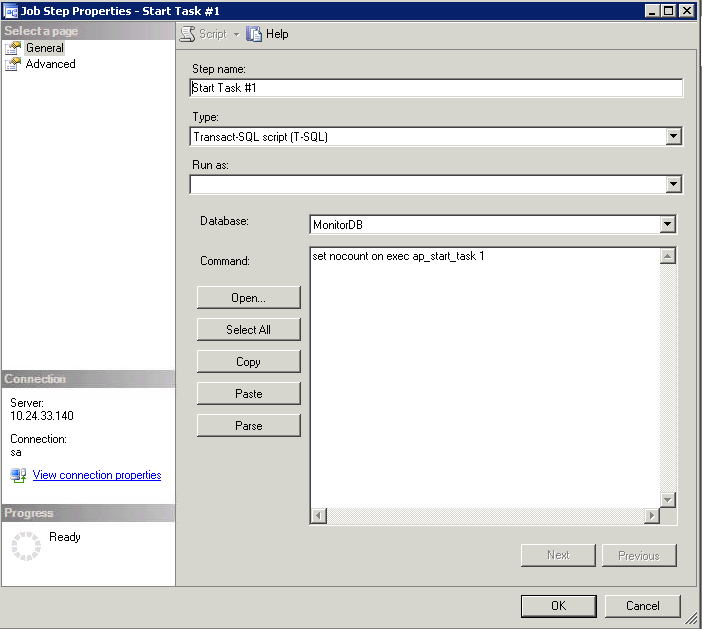
Actualmente puede mostrar hasta 7 imágenes por directorio, Climax enviará un máximo de 6, si hubiese que agregar por cambio de definición de Climax, , se configura desde el style.css. en la línea 31 y 73. Si llegase un directorio que está vacío, mostrará con letras rojas en etiqueta H1 que el directorio se encuentra vacío.

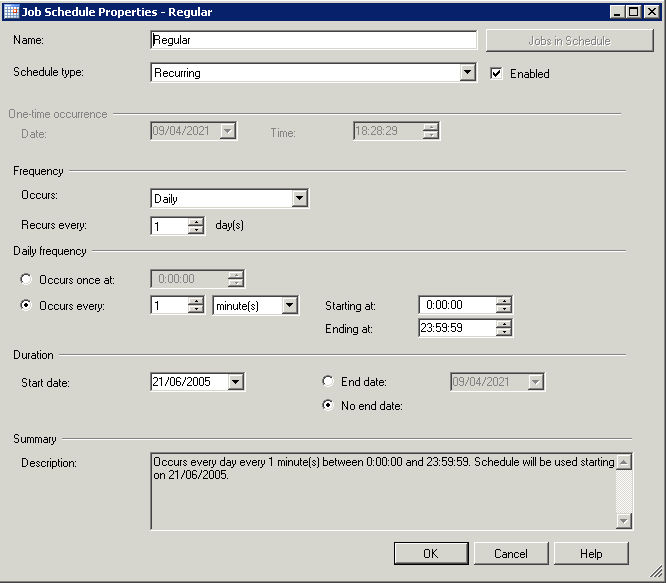
**3\_Job.Sql**

Será el job que ejecute ap\_climax cada un minuto si es que no está ejecutándose, análogo al procesamiento de señales de la task 101 con el sp ap\_o configurado en el task status en el campo start command.

Es importante que el job se esté ejecutando apuntando a la tarea correcta, caso contario no funcionará.

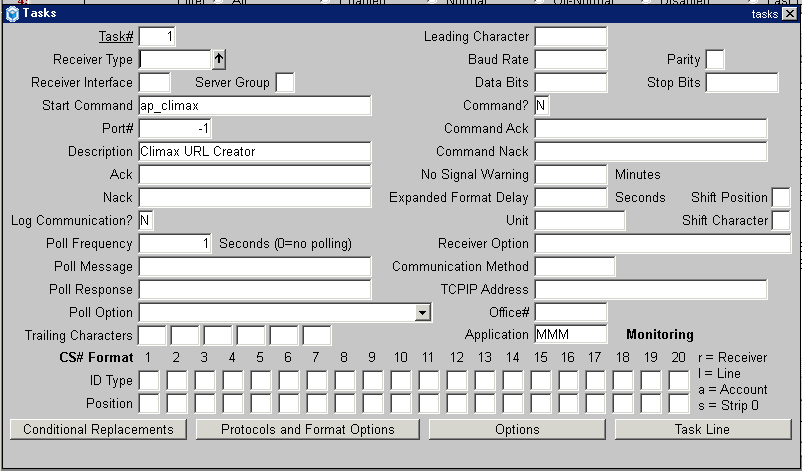
Ejemplo de configuración del job que apunta a la task 1 del entorno de test 10.24.33.140 monitordb





**6\_ Tarea en MasterMind**

Se configura de la siguiente manera.



**CAMBIOS RECIENTES:**

**VERSION 1.2**

**Se agrega autocomplete de la tabla OPTION y de los task\_option de la task, mostrando error en la ventana task\_status si lo hubiese para así colaborar en trobleshooting.**

**Se agrega funcionalidad de trobleshooting extra:**

**Si se ejecuta exec ap\_climax -task\_no (Notese el signo menos delante de la task a probar)**

**Resultado en caso de ok:**

VERSION 1.2

NADA PARA SETEAR EN OPTIONS ESTÁN SUS 9 VARIABLES

@CLIMAX\_DELAY\_TIME 00:00:05:000

@CLIMAX\_HTTP\_SERVER\_URL http://10.24.34.23/index.php?uri=

@CLIMAX\_MAS\_PREFIX VF|MAS|

@CLIMAX\_MAX\_ROW\_PROCESSING 200

@CLIMAX\_START\_URL\_TAG <LINK>

@CLIMAX\_END\_URL\_TAG <LINK/>

@CLIMAX\_MINUTES\_BEFORE 60

@CLIMAX\_IMAGE\_EVENTS <event\_id>CXLINK</event\_id>

@CLIMAX\_TASK\_MONITORING <task\_no>29</task\_no><task\_no>229</task\_no><task\_no>284</task\_no>

**Ejemplos.**

**Al arrancar el programa, comprueba si existen los 9 OPTIONS EN MASTERMIND, en caso de no existir, los agrega:**

INSERT INTO OPTIONS VALUES

('CLIMAX\_DELAY\_TIME','CLIMAX\_DELAY\_TIME',1,GETDATE(),'N',NULL,NULL,NULL,'N','T',NULL,'N','Y'),

('CLIMAX\_END\_URL\_TAG','CLIMAX\_END\_URL\_TAG',1,GETDATE(),'N',NULL,NULL,NULL,'N','T',NULL,'N','Y'),

('CLIMAX\_HTTP\_SERVER\_URL','CLIMAX\_HTTP\_SERVER\_URL',1,GETDATE(),'N',NULL,NULL,NULL,'N','T',NULL,'N','Y'),

('CLIMAX\_IMAGE\_EVENTS','CLIMAX\_IMAGE\_EVENTS',1,GETDATE(),'N',NULL,NULL,NULL,'N','T',NULL,'N','Y'),

('CLIMAX\_MAS\_PREFIX','CLIMAX\_MAS\_PREFIX',1,GETDATE(),'N',NULL,NULL,NULL,'N','T',NULL,'N','Y'),

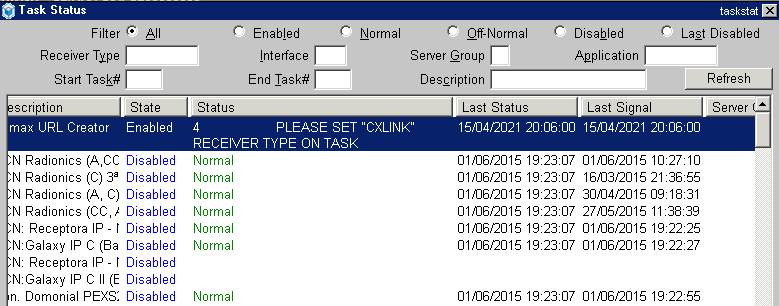
('CLIMAX\_MAX\_ROW\_PROCESSING','CLIMAX\_MAX\_ROW\_PROCESSING',1,GETDATE(),'N',NULL,NULL,'N','N','T',NULL,'N','Y'),

('CLIMAX\_MINUTES\_BEFORE','CLIMAX\_MINUTES\_BEFORE',1,GETDATE(),'N',NULL,NULL,NULL,'N','T',NULL,'N','Y'),

('CLIMAX\_START\_URL\_TAG','CLIMAX\_START\_URL\_TAG',1,GETDATE(),'N',NULL,NULL,NULL,'N','T',NULL,'N','Y'),

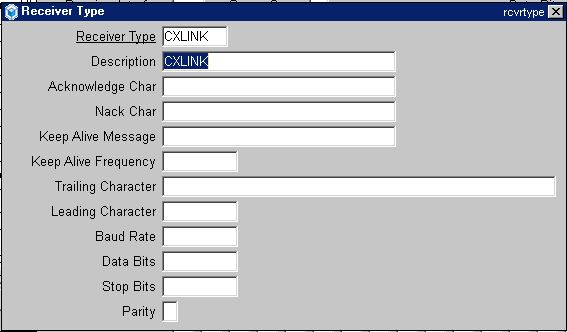
('CLIMAX\_TASK\_MONITORING','CLIMAX\_TASK\_MONITORING',1,GETDATE(),'N',NULL,NULL,NULL,'N','T',NULL,'N','Y')

**Si no está configurado el receiver type CXLINK en la TASK, se mostrará lo siguiente:**

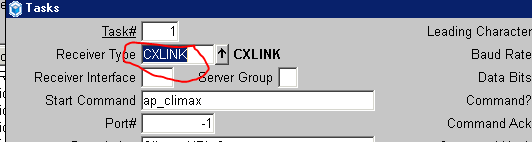


**Solución :**

**1\_ Crear el receiver type**

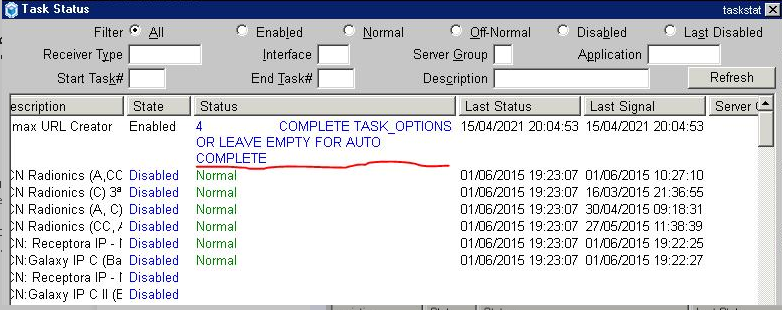


**2\_Colocarlo en la task**



**Reiniciar la task.**

**2\_ En el caso de que falte algún task\_option saldrá el siguiente mensaje.**



**Solución:**

**Complete los 9 task\_option por sql o por GUI de master, si es por sql es:**

insert into m\_task\_option values

(@thisTask,'CLIMAX\_DELAY\_TIME',getdate(),1,'00:00:05:000'),

(@thisTask,'CLIMAX\_HTTP\_SERVER\_URL',getdate(),1,'http://10.24.34.23/index.php?uri='),

(@thisTask,'CLIMAX\_IMAGE\_EVENTS',getdate(),1,'CXLINK'),

(@thisTask,'CLIMAX\_MAS\_PREFIX',getdate(),1,'VF|MAS|'),

(@thisTask,'CLIMAX\_MAX\_ROW\_PROCESSING',getdate(),1,'200'),

(@thisTask,'CLIMAX\_MINUTES\_BEFORE',getdate(),1,'60'),

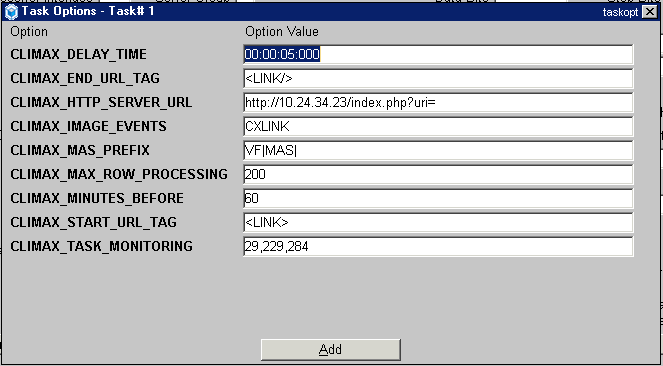
(@thisTask,'CLIMAX\_START\_URL\_TAG',getdate(),1,'<LINK>'),

(@thisTask,'CLIMAX\_END\_URL\_TAG',getdate(),1,'<LINK/>'),

(@thisTask,'CLIMAX\_TASK\_MONITORING',getdate(),1,'29,229,284')

**O mas sencillo, deje la task sin task\_options, se autocompletará con los valores por default arriba descriptos.**

**Así debería quedar:**



**Aclaración sobre el option CLIMAX\_HTTP\_SERVER\_URL.**

**Es importante que termine en ?uri= Luego de la url donde se encuentra el index.php (VisorSpider)**

**Ya que URI, es el nombre de la variable dentro del php, que recibirá una url para abrir y mostrar al usuario.**

VEA EL CODE COMPLETO EN https://github.com/