

# ¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.  
 Ese apoyo que siempre quiso tener...

## 1. Desarrollo de componentes y proyectos a medida



## 2. Auditoría de código y recomendaciones de mejora

## 3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



## 4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces,  
 HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)  
 Gestor de contenidos (Alfresco)  
 Aplicaciones híbridas

Tareas programadas (Quartz)  
 Gestor documental (Alfresco)  
 Inversión de control (Spring)

Control de autenticación y  
 acceso (Spring Security)  
 UDDI  
 Web Services  
 Rest Services  
 Social SSO  
 SSO (Cas)

JPA-Hibernate, MyBatis  
 Motor de búsqueda empresarial (Solr)  
 ETL (Talend)

Dirección de Proyectos Informáticos.  
 Metodologías ágiles  
 Patrones de diseño  
 TDD

BPM (jBPM o Bonita)  
 Generación de informes (JasperReport)  
 ESB (Open ESB)

Estás en:

[Inicio](#) [Tutoriales](#) Notificación de eventos en Nut



DESARROLLADO POR:  
 Ángel García Jerez

Consultor tecnológico de desarrollo de proyectos informáticos. Co-autor del libro "Actualización y mantenimiento del PC (Edición de 2010) publicado por Anaya Multimedia

Ingeniero Técnico en Informática de Sistemas e Ingeniero en Informática (premio al mejor expediente de su promoción)

Puedes encontrarme en Autentia: Ofrecemos servicios de soporte a desarrollo, factoría y formación

Somos expertos en Java/J2EE

Catálogo de servicios Autentia



Últimas Noticias

XIV Charla Autentia - ZK

Java on Line profesional: Las reglas no escritas para triunfar en la empresa. 2ª EDICIÓN ACTUALIZADA.

Pequeño coding dojo con Carlos Ble en las oficinas de Autentia.

Disponible gratis, Autentia Comic para el iPhone y iPad,

Comentando #AID2010. Agil Industrial Day 30 Nov 2010

Histórico de NOTICIAS

Últimos Tutoriales

Cosas que no funcionan en JSF2 como uno podría esperar (es decir, bugs)

Ejemplo de Swing Worker: ¿Por qué se me congela la interfaz?

Nut - Network UPS Tools

Anuncios Google [Java](#) [E Finance Java](#) [Java Code Sign](#)

Fecha de publicación del tutorial: 2011-01-31



Share |

[Regístrate para votar](#)

## NOTIFICACIÓN DE EVENTOS EN NUT

1. Introducción
2. Entorno
3. Activando la notificación de eventos en upsmon
4. Configuración
5. Conclusiones.

### 1. Introducción

En un primer tutorial de Nut (pichar [aquí](#)) vimos su instalación y configuración básica. Ahora veremos algo un poco más avanzado, configurar upsmon para que nos notifique ante eventos producidos por nuestra UPS o SAI.

Upsmon (servicio encargado de monitorizar la UPS) proporciona entre sus características un sistema de notificación de eventos ante diversas situaciones producidas en la UPS. Perdida/recuperación de corriente eléctrica, perdida/establecimiento de comunicación con la UPS, baja batería son algunos de ellos permitiéndonos estar totalmente informado ante un cambio de estado de nuestro sistema.

Por defecto, upsmon, notifica los eventos a través de: syslog (SYSLOG) y mensajes de consola a todos los usuarios del sistema (WALL) pero este comportamiento puede ser modificado a través de /etc/nut/upsmon.conf. Podremos individualizar el tipo de notificación de cada evento, es decir, puede que ciertos eventos sólo nos interesen que se registren en el syslog y que otros nos envíe un mensaje en tiempo real por email o SMS. Esto último podemos hacerlo gracias a que upsmon tiene, por decirlo de

alguna manera, la posibilidad de añadir nuevas formas de notificación.

¿Cómo lo hace? Ejecutando un comando que previamente hemos configurado. Éste será invocado con un único parámetro que representa el mensaje del evento producido y dos variables de entorno: NOTIFYTYPE y UPSNAME que corresponde con el tipo de evento y el nombre de la UPS que ha generado el evento.

Aunque podríamos crear nuestro propio programa que se comportase o realizase determinadas tareas ante cambios de estado de nuestra UPS, Nut, ofrece ya una utilidad bastante potente al respecto llamada upssched (fichero de configuración /etc/nut/upssched.conf). Podremos utilizar timers, para que no se notifique de un evento si éste se ha restablecido en un periodo de tiempo. Va a quedar claro con un ejemplo; imaginaos que se va la luz y por tanto nuestra UPS entra en modo batería pero se restablece al cabo de los 5 segundos. Con los timers podemos indicar que no se notifique de ningún cambio de estado siempre que no perdure más de un intervalo de tiempo.

## 2. Entorno

Entorno utilizado para escribir este tutorial:

- **Sistema Operativo:** Ubuntu Server 64 bits
- **Kernel:** 2.6.35-22
- **nut:** 2.4.3
- **UPS:** Trust 1300VA

## 3. Activando la notificación de eventos en upsmon

Upsmon es capaz de notificar diez tipos de evento:

- **ONLINE:** recuperación de corriente eléctrica.
- **ONBATT:** pérdida de corriente eléctrica.
- **LOWBATT:** batería baja.
- **FSD:** apagado forzoso del sistema.
- **COMMOK:** comunicación establecida con la UPS.
- **COMMBAD:** comunicación perdida con la UPS.
- **SHUTDOWN:** el sistema se está apagando.
- **REPLBATT:** la batería debe ser cambiada.
- **NOCOMM:** no se puede conectar a la UPS para la monitorización.
- **NOPARENT:** el proceso que se ha lanzado para el apagado del sistema ha fallado.

En este tutorial vamos a configurar upsmon/upssched para que nos notifique por email de los eventos:

- pérdida/recuperación de luz (ONBATT y ONLINE).
- establecimiento/pérdida de comunicación con la UPS (COMMOK y COMMBAD).

Lo primero que tenemos que hacer es indicar el comando encargado de procesar los eventos producidos por la UPS. En nuestro caso utilizaremos upssched.

```
NOTIFYCMD /sbin/upssched
```

A continuación, establecemos el comportamiento que debe tener upsmon ante los diferentes eventos producidos. Los posibles valores son SYSLOG, WALL o EXEC que representan notificación a través del syslog, envío de un mensaje de consola a todos los usuarios o ejecución a través del comando establecido en la variable NOTIFYCMD, respectivamente.

-  Awstats - Herramienta de generación de estadísticas.
-  Utilización de Commons Digester para un sistema de preferencias configurable

Últimos Tutoriales del Autor

-  Nut - Network UPS Tools
-  Awstats - Herramienta de generación de estadísticas.
-  JBoss autenticación basada en certificados cliente
-  Rdiff-backup: Herramienta para realizar backups
-  ACLExtension: Permisos de grano fino en Mercurial

Síguenos a través de:



Últimas ofertas de empleo

- 2010-10-11  Comercial - Ventas - SEVILLA.
- 2010-08-30  Otras - Electricidad - BARCELONA.
- 2010-08-24  Otras Sin catalogar - LUGO.
- 2010-06-25  T. Información - Analista / Programador - BARCELONA.

NOTIFYFLAG ONLINE	SYSLOG+WALL+EXEC
NOTIFYFLAG ONBATT	SYSLOG+WALL+EXEC
NOTIFYFLAG LOWBATT	SYSLOG+WALL+EXEC
NOTIFYFLAG FSD	YSLOG+WALL
NOTIFYFLAG COMMOK	SYSLOG+WALL+EXEC
NOTIFYFLAG COMMBAD	SYSLOG+WALL+EXEC
NOTIFYFLAG SHUTDOWN	SYSLOG+WALL
NOTIFYFLAG REPLBATT	SYSLOG+WALL+EXEC
NOTIFYFLAG NOCOMM	SYSLOG+WALL
NOTIFYFLAG NOPARENT	SYSLOG+WALL

Ahora sólo nos queda configurar upsched. Upsshed utiliza un fichero el cuál será realmente el responsable de procesar los eventos y realizar la tarea en concreto (que en nuestro caso es la de enviar un email). Pero podéis pensar por qué no configuramos nuestro fichero en upsmon y así evitamos utilizar upssched que únicamente es una especie de proxy. Pues realmente es porque el verdadero potencial de upssched es el uso de timers, es decir, tendremos la posibilidad de que si un evento no perdura más de un intervalo de tiempo upssched no ejecuta nuestro fichero de procesamiento de eventos.

Además de los TIMERS también tenemos la posibilidad de que cuando se produzca un evento se notifique inmediatamente del hecho usando el comando EXECUTE.

La nomenclatura que tendremos que utilizar en upssched es muy sencilla. Para los timers tenemos un comando para que comience el evento:

```
AT TIPO_EVENTO NOMBRE_UPS START-TIMER NOMBRE_EVENTO
INTERVALO_DE_TIEMPO
```

Y otro para inhabilitar el timer:

```
AT TIPO_EVENTO NOMBRE_UPS CANCEL-TIMER NOMBRE_EVENTO
```

Por ejemplo, las líneas que tenemos que añadir en el caso de que si se recupera la luz antes de 10 segundos no se notifique de la perdida de esta sería:

```
AT ONBATT * START-TIMER ups-on-battery 10
AT ONLINE * CANCEL-TIMER ups-on-battery
```

En el caso de que utilizemos los timers tenemos que tener en cuenta que debemos configurar las variables PIPEFN y LOCKFN para que funcionen.

Para la ejecución inmediata es algo similar:

```
AT ONLINE * EXECUTE ups-back-on-line
```

A continuación os nuestro un fichero de ejemplo:

```
01. CMDSCRIPT /var/adictos/ups/ups-send-mail.sh
02. PIPEFN /var/run/nut/upssched/upssched.pipe
03. LOCKFN /var/run/nut/upssched/upssched.lock
04. AT ONBATT * START-TIMER ups-on-battery 10
05. AT COMBAD * START-TIMER ups-comunication-bad 10
06. AT ONLINE * CANCEL-TIMER ups-on-battery
07. AT COMOK * CANCEL-TIMER ups-comunication-bad
08.
09. AT REPLBATT * EXECUTE ups-change_battery
10. AT ONLINE * EXECUTE ups-back-on-line
11. AT LOWBATT * EXECUTE ups-low-battery
12. AT COMOK * EXECUTE ups-comunication-ok
```

Como habéis podido ver en la línea 1 cuando upssched tiene que notificar de un evento ejecutará el comando ups-send-mail.sh. Éste script recibirá como primer parámetro el nombre del evento configurado en upssched, es decir, on-battery, on-back-on-line, etc...

```

01. #!/bin/bash
02.
03. UPS_SCRIPT_DIR=/var/adictos/ups
04. MAIL_DEST=""
05. FSFREE_FILE="/tmp/fsfree.txt"
06.
07. MESSAGE_MAIL=""
08. EVENT_TYPE=$1
09.
10. case "$EVENT_TYPE" in
11.     ups-on-battery)
12.         MESSAGE_MAIL="UPS on battery"
13.         ;;
14.     ups-comunication-bad)
15.         MESSAGE_MAIL="Communications with UPS lost"
16.         ;;
17.     ups-change_battery)
18.         MESSAGE_MAIL="UPS battery needs to be replaced"
19.         ;;
20.     ups-back-on-line)
21.         MESSAGE_MAIL="UPS on line power"
22.         ;;
23.     ups-low-battery)
24.         MESSAGE_MAIL="UPS battery is low"
25.         ;;
26.     ups-comunication-ok)
27.         MESSAGE_MAIL="Communications with UPS established"
28.         ;;
29. esac
30.
31. echo "$MESSAGE_MAIL" | mail -s "[UPS] Event $EVENT_TYPE de `uname -n`: " c

```

## 4 Conclusión

Como habéis podido ver la notificación de eventos con Nut es bastante sencilla de configurar. Desde [Autentia](#) siempre intentamos publicar tutoriales que os puedan ayudar a mejorar vuestro entorno de trabajo e intentar minimizar al máximos posibles problemas que puedan surgir en el mantenimiento de nuestro sistemas.

Anímate y coméntanos lo que pienses sobre este **TUTORIAL**:

Puedes opinar o comentar cualquier sugerencia que quieras comunicarnos sobre este tutorial; con tu ayuda, podemos ofrecerte un mejor servicio.

Enviar comentario

(Sólo para usuarios registrados)

» [Regístrate](#) y accede a esta y otras ventajas «

## COMENTARIOS



Esta obra está licenciada bajo licencia Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5