

Lycée Sacré Cœur



NUT et La Gestion des Onduleurs

Réalisé par Nathalie Desmares

Table des matières

I. <u>Les onduleurs</u>	4
Présentation.....	4
Les types d'onduleur.....	4
Caractéristiques de l'onduleur.....	4
II. <u>Debian</u>	5
III. <u>Contexte du projet</u>	5
IV. <u>Présentation de NUT</u>	6
V. <u>Installation de NUT sur Debian</u>	6
VI. <u>Configuration de NUT en mode Master (Maître)</u>	7
VI.1. NUT.....	7
VI.2. Donner les droits à l'utilisateur NUT pour l'accès au port série.....	7
VI.3. Communiquer avec l'UPS.....	7
VI.4. Le pilote de l'onduleur.....	8
VI.5. Configuration du demon upsd.....	9
VI.6. Définition des utilisateurs.....	9
VI.7. Les droits d'accès aux fichiers de configuration.....	9
VI.8. Paramétrage de la surveillance de l'UPS.....	9
VI.9. Lancement de NUT.....	10
VI.10. Le délai au bout duquel le serveur va s'éteindre.....	10
VI.10.1. Le paramétrage du demon upsmon.....	11
VI.10.2. Le paramétrage de « Upssched ».....	11
VI.11. Tester la configuration.....	12
VI.12. La programmation de l'UPS.....	13
VII. <u>Configuration de NUT en mode Esclave</u>	14
VII.1. Installation de NUT.....	14
VII.2. Configuration du fichier <code>/etc/default/nut</code>	14
VII.3. Configuration du demon upsmon.....	14
VIII. <u>Configuration de NUT sur Windows (Esclave)</u>	15
VIII.1. Installation de WinNUT.....	15
VIII.2. Configuration de WinNUT.....	15
VIII.3. Paramétrage du fichier upsmon.....	15
IX. <u>Les outils de surveillance</u>	16
IX.1. Configuration du demon SNMP.....	16



IX.1.1. Principe de fonctionnement du protocole SNMP.....	16
IX.1.2. Installation	16
IX.1.3. Configuration	16
IX.1.4. Redémarrage du demon snmpd	17
IX. 2. L'interface web	17
IX.3. Configuration de WinNUTClient.....	18
IX.4. PRTG.....	19



I. Les onduleurs

PRESENTATION

Un **onduleur** (en anglais **UPS** pour *Uninterruptible Power Supply*) est un dispositif permettant de protéger des matériels électroniques contre les aléas électriques. Il s'agit ainsi d'un boîtier placé en interface entre le réseau électrique (branché sur le secteur) et les matériels à protéger.

L'onduleur permet de basculer sur une batterie de secours pendant quelques minutes en cas de problème électrique, notamment lors de :

- Micro-coupures de courant, c'est-à-dire des coupures électriques de quelques millièmes de seconde pouvant se traduire par le redémarrage de l'ordinateur.
- Coupure électrique, correspondant à une rupture en alimentation électrique pendant un temps déterminé.
- Surtension, c'est-à-dire une valeur nominale supérieure à la valeur maximale prévue pour le fonctionnement normal des appareils électriques.
- Sous-tension, c'est-à-dire une valeur nominale inférieure à la valeur maximale prévue pour le fonctionnement normal des appareils électriques.
- Pics de tension. Il s'agit de surtensions instantanées (pendant un temps très court) et de forte amplitude. Ces pics, dûs à l'arrêt ou la mise en route d'appareils de forte puissance, peuvent à terme endommager les composants électriques.
- Foudre, représentant une surtension très importante intervenant brusquement lors d'intempéries (orages).

La majeure partie des perturbations électriques sont tolérées par les systèmes informatiques, mais elles peuvent parfois causer des pertes de données et des interruptions de service, voire des dégâts matériels.

L'onduleur permet de « lisser » la tension électrique, c'est-à-dire supprimer les crêtes dépassant un certain seuil. En cas de coupure de courant, l'énergie emmagasinée dans la batterie de secours permet de maintenir l'alimentation des matériels pendant un temps réduit (de l'ordre à 5 à 10 minutes). Au-delà du temps d'autonomie de la batterie de l'onduleur, celui-ci permet de basculer vers d'autres sources d'énergie. Certains onduleurs peuvent également être branchés directement à l'ordinateur (via un câble [USB](#) par exemple) afin de commander proprement son extinction en cas de panne de courant et éviter toute perte de données.

LES TYPES D'ONDULEUR

On distingue généralement trois familles d'onduleurs :

- **Les onduleurs dits « Off-Line »** sont branchés via un relais électrique. En fonctionnement normal, la tension du réseau électrique sert à recharger les batteries. Lorsque la tension passe au-delà d'un certain seuil (minimal et maximal), le relais s'ouvre et la tension est recrée à partir de l'énergie emmagasinée dans la batterie. Compte-tenu des temps d'ouverture et de fermeture du relais, ce type d'onduleur ne permet pas une protection contre les micro-coupures.
- **Les onduleurs dits « On-Line »** sont branchés en série et régulent en permanence la tension.
- **Les onduleurs dits « Line interactive »** sont issus d'une technologie hybride. En effet, les onduleurs *line interactive* sont branchés en parallèle via un relais mais possèdent un micro-processeur surveillant la tension électrique en continu. En cas de faible baisse de tension ou de micro-coupures, l'onduleur est capable d'injecter une tension compensatoire. En cas de panne totale par contre, l'onduleur fonctionnera comme un onduleur offline.

Un comparatif est disponible en annexe 1 page

CARACTERISTIQUES DE L'ONDULEUR

La durée de la protection électrique fournie par un onduleur est exprimée en **VA** (*Volt-Ampère*). On considère généralement que pour une protection électrique correspondant à une coupure électrique de 10 minutes, il est nécessaire de se doter d'un onduleur ayant une capacité égale à la puissance de l'ensemble des matériels raccordés à l'onduleur multiplié par un coefficient 1,6.

Lors du choix d'un onduleur, il est également important de vérifier le nombre de prises électriques qu'il possède.



Enfin, les onduleurs possèdent parfois une connectique ([USB](#), réseau, parallèle, etc.) permettant de les relier à l'unité centrale, afin de permettre de commander son extinction en cas de panne de courant prolongée et ainsi sauvegarder l'ensemble des travaux en cours.

Il est à noter également que les onduleurs ne protègent pas les connexions téléphoniques, ainsi un ordinateur connecté à un onduleur ainsi qu'à un modem risque de voir ses composants endommagés si la foudre frappe la ligne téléphonique.

II. Debian

Le logiciel NUT s'imposant dans la gestion des onduleurs et fonctionnant sur une distribution Linux, le choix du système d'exploitation c'est donc porté sur une Linux Debian Etch, la dernière version stable.

Beaucoup d'établissements fonctionnent encore sur la distribution Linux Debian Sarge. L'ensemble du projet sera donc validé également pour celle-ci.

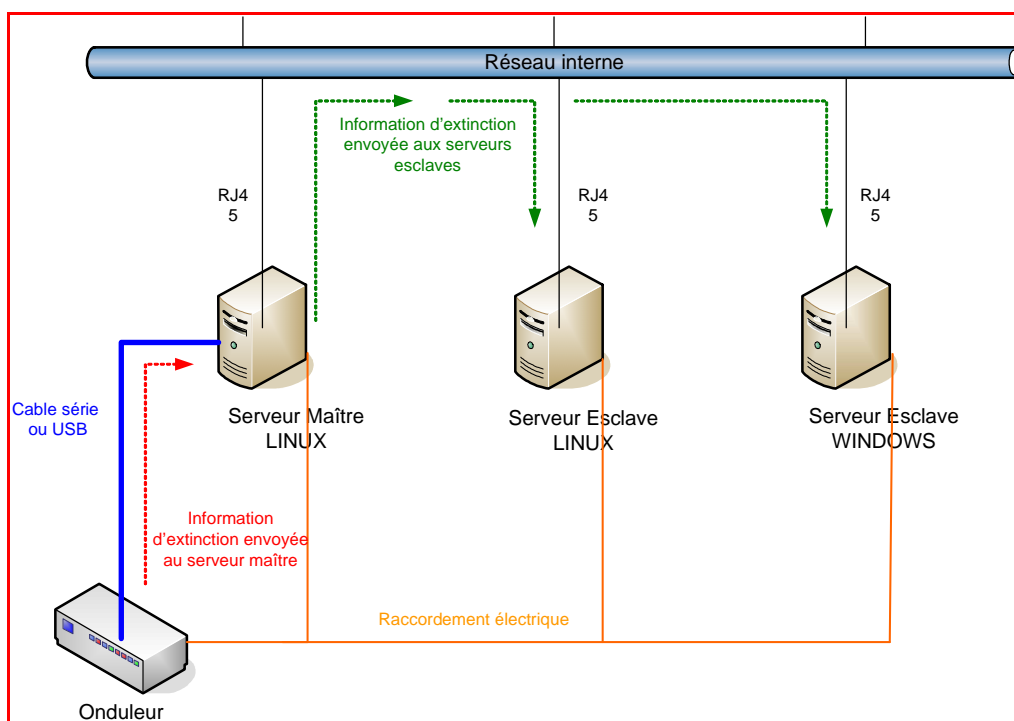
La procédure d'installation de la distribution Linux Debian Etch est disponible en annexe ___

Attention :

Sarge étant une ancienne version, elle est passée en archives. Il faut donc modifier le fichier `/etc/sources.list` avant de pouvoir mettre à jour le système.

La source à préciser : **deb <http://archive.debian.org/debian> sarge main contrib non-free**

III. Contexte du projet



IV. Présentation de NUT

Network UPS tool permet de surveiller l'alimentation électrique d'un onduleur par le port série de l'ordinateur; il provoque l'extinction propre du système lorsque la panne de courant a été assez longue pour faire descendre la charge des batteries à un état critique ; ainsi les partitions des disques durs sont démontées normalement et l'ensemble de leur contenu, système comme données, ne court aucun risque. En effet, une rupture brutale d'alimentation d'un ordinateur peut avoir des conséquences catastrophiques sur le contenu des disques : onduler un serveur ne suffit pas, encore faut-il surveiller cet onduleur avec un logiciel qui sait ce qu'est une rupture d'alimentation et comment y faire face.

Nut comporte également une partie cliente qui lui permet de fonctionner en réseau, c'est à dire interroger par TCP/IP un autre serveur qui surveille par sa liaison série l'onduleur, qui ne dispose que d'un seul port pour cela. Ce ou ces serveurs esclaves tomberont les premiers, avant le maître, en cas de baisse critique de l'alimentation, le maître attendant la chute des esclaves avant de s'éteindre. La richesse fonctionnelle de Nut en fait un des logiciels les plus puissants pour ce type d'usage.

On respectera trois démarches de bon sens lors de l'installation d'onduleurs et de leur surveillance :

1. Tout le matériel actif doit être ondulé, en particulier les switchs qui relient les serveurs partageant le même onduleur. Comme l'esclave surveille l'onduleur du maître par réseau, cette surveillance fonctionnera moins bien lors d'une panne si le switch n'est pas ondulé. Si l'esclave surveille le maître en l'appelant par son nom et non par son adresse, on veillera à la disponibilité du service DNS en cas de coupure, et donc faire tourner celui ci sur le serveur maître.
2. Dans la configuration du logiciel de surveillance, on laissera les batteries de l'onduleur se vider jusqu'à une charge critique avant de faire tomber le serveur, au lieu de le faire à la moindre coupure : c'est la fonction même d'un onduleur.
3. De nombreux tests seront nécessaires pour vérifier que tout fonctionne bien, en particulier le retour des serveurs en fonction suite au rétablissement de l'alimentation.

Enfin avant toute installation, vérifier la compatibilité du matériel.

La liste des drivers et des matériels compatibles à NUT est accessible sur le site : <http://eu1.networkupstools.org>
(onglet : Compatibility)

La procédure qui suit à été testée et validée sur différents matériels et 2 Versions de Débian.

Sur Debian Etch noyau 2.6., les matériels suivant ont été testés :

Onduleur APC CyberFort 350
Onduleur Infosec Ipel Evolution
Onduleur Belkin Universel F6C800exUNV

Sur Debian Sarge noyau 2.6., les matériels suivant ont été testés :

Onduleur Belkin Universel F6C800exUNV
Onduleur Infosec Ipel Evolution

V. Installation de NUT sur Debian

Dans un premier temps, mettre à jour la liste des sources utilisée par Debian en utilisant la commande :

```
# apt-get update
```

Ensuite, l'installation de NUT se fait par la commande:



```
# apt-get install nut
# apt-get install nut-usb
```

VI. Configuration de NUT en mode Master (Maître)

Nut se comporte en mode client/serveur même sur un seul ordinateur : le serveur s'appelle upsd, c'est lui qui obtient les informations sur l'état des batteries par le port série. Le client, lui, s'appelle upsmon, il s'authentifie auprès d'upsd, surveille l'onduleur et provoque l'arrêt du serveur, puis de l'onduleur lui-même.

En principe, un répertoire **/etc/nut** est créé, vide de tout contenu.

Il contiendra par la suite 6 fichiers :

- **ups.conf** : l'onduleur lui-même
- **upsd.conf** : le démon qui surveille l'onduleur par le câble
- **upsd.users** : les utilisateurs du démon et leur authentification
- **upsmon.conf** : le client qui, une fois authentifié, provoquera les actions sur le serveur et l'onduleur en fonction des informations que lui communique le démon
- **upsshed.conf** : « upsshed » étant l'utilitaire qui a été ajouté au package de NUT pour gérer l'exécution de scripts en différé.
- **upsalert** : script permettant la gestion des messages d'alertes et des arrêts différés.

VI.1. NUT

NUT est composé de 2 démons :

- upsd qui surveille l'UPS et fournit les informations à upsmon
- upsmon. qui est le programme réseau qui se connecte au serveur (soit lui-même si on est sur la machine maître, soit distant si on est sur une machine esclave) pour connaître l'état de l'UPS et lancer les actions adéquates sur le système

Lors du démarrage de NUT par le script Init-V **/etc/init.d/nut start**, il faut lui préciser quels démons démarrer.

Pour cela, écrire dans le fichier **/etc/default/nut** :

```
# start upsd
START_UPSD=yes

# start upsmon
START_UPSMON=yes
```

VI.2. DONNER LES DROITS A L'UTILISATEUR NUT POUR L'ACCES AU PORT SERIE

Les droits attribués à NUT sont donnés par ligne de commande :

```
# chmod 0600 /dev/ttyS0
# chown nut:nut /dev/ttyS0
# addgroup nut dialout
```

ttyS0 est un port série (com1)(S0 c'est S "zéro")

La dernière ligne permet à NUT de faire partie de dialout, qui est autorisé par udev (gestionnaire de périphériques) à accéder à ttyS0.

VI.3. COMMUNIQUER AVEC L'UPS.



Pour communiquer avec l'UPS, il faut choisir un driver et un port de communication adéquats.

Ecrire dans le fichier */etc/nut/ups.conf*:

Pour un UPS connecté sur le port série :

```
[ipel]
driver = megatec #(ou esupssmart pour la Sarge)
port = /dev/ttyS0
desc = "ipel"
```

Pour un UPS connecté sur un port USB

```
[nom_ups]
driver= newhidups
port= auto
```

Entre crochets est défini le nom donné à l'UPS, ici "nom_UPS". Ceci permet de l'identifier car NUT peut gérer plusieurs UPS.

Puis, préciser le pilote qui est "newhidups" pour les UPS USB

Rappel : La liste des drivers et des matériels compatibles à NUT est accessible sur le site :

<http://eu1.networkupstools.org/compat/stable.html>

VI.4. LE PILOTE DE L'ONDULEUR

Avant c'est deux opérations, s'assurer que NUT ne soit pas en service.

```
/etc/init.d/# sh nut stop
```

Tester, sous root, le bon fonctionnement du pilote de l'onduleur à travers le câble série. Les pilotes se trouvent dans /lib/nut, la commande appropriée est:

```
# /lib/nut/megatec /dev/ttyS0
```

résultat :

```
Etch:/etc/init.d# upsdrcctl start
Network UPS Tools - UPS driver controller 2.0.4
Network UPS Tools - Megatec protocol driver 1.4 (2.0.4)
Carlos Rodrigues (c) 2003-2006
```

```
Megatec protocol UPS detected.
```

Le démarrage de **upsdrvctl** permettra de vérifier que l'onduleur est bien connecté et reconnu.

```
# upsdrcctl start
```

résultat à obtenir :

```
Etch:/etc/init.d# /lib/nut/megatec /dev/ttyS0
Network UPS Tools - Megatec protocol driver 1.4 (2.0.4)
Carlos Rodrigues (c) 2003-2006
```

```
/dev/ttyS0 is locked by another process
```



VI.5. CONFIGURATION DU DEMON UPSD

La configuration du démon réseau au niveau des accès se fait via le fichier `/etc/nut/upsd.conf` :

```
# Liste des ACLs:  
ACL all 0.0.0.0/  
ACL sacrecoeur 192.168.40.0/24  
ACL localhost 127.0.0.1/32  
ACL nut-cgi 10.36.0.10/32  
  
# les droits:  
ACCEPT localhost sacrecoeur nut-cgi  
REJECT all
```

Ce fichier définit les access lists (ACL) et les droits de chacune.
Seuls les accès depuis le poste local et le réseau local.

VI.6 DEFINITION DES UTILISATEURS

Cette étape consiste à définir un utilisateur avec un mot de passe qui pourra accéder au démon par le fichier `/etc/nut/upsd.users` :

```
[admin]  
password = adminpw  
allowfrom = localhost  
actions = SET  
instcmds = ALL  
upsmon master
```

"upsmon master" précise que cet utilisateur a le droit de se connecter sur le serveur maître pour contrôler l'UPS.

VI.7. LES DROITS D'ACCES AUX FICHIERS DE CONFIGURATION

Il faut définir les droits sur les différents fichiers de configuration :

```
# chown -R root.nut /etc/nut  
# chmod 755 /etc/nut  
# chmod 640 /etc/nut/*  
# chmod 644 /etc/nut/hosts.conf
```

VI.8. PARAMETRAGE DE LA SURVEILLANCE DE L'UPS

C'est le client le plus important de la « collection » NUT. Son rôle est de :

- lancer par le script de démarrage.
- surveiller l'état de l'onduleur.
- arrêter l'ordinateur avant épuisement des batteries.
- communiquer directement avec le démon.

Il se configure à partir du fichier `/etc/nut/upsmon.conf`.

```
# définition des UPS à surveiller et des autorisations  
MONITOR ipel@localhost 1 admin adminpw master
```



MINSUPPLIES 1

définition de la commande Shutdown
SHUTDOWNCMD "/sbin/shutdown -h +0"

POLLFREQ 5
POLLFREQALERT 5
HOSTSYNC 15
DEADTIME 15
POWERDOWNFLAG /etc/killpower
FINALDELAY 5

Il contient l'utilisateur/mot de passe qui sera utilisé pour la surveillance de l'onduleur ainsi que la commande a exécutée lorsque l'onduleur devra éteindre la machine ("SHUTDOWNCMD").ainsi que d'autres directives :

- POLLFREQ : fréquence d'interrogation du démon UPSD.
 - POLLFREQALERT : fréquence d'interrogation du démon UPSD lorsque l'onduleur est sur batterie.
 - DEADTIME : la durée en seconde à partir de laquelle l'onduleur est considéré comme « mort ».
 - SHUTDOWNCMD : commande utilisée pour arrêter le système.
 - NOTIFYCMD : commande utilisée pour émettre des notifications.
 - NOTIFYFLAG : associe à un événement des méthodes d'envoi de notification :
NOTIFYFLAG <notify type> <flag> ...
 - SYSLOG : écriture d'une trace dans le fichier syslog.
 - WALL :: envoi d'un message sur la console des utilisateurs connectés au système (commande wall).
 - EXEC : exécute la commande définie par NOTIFYCMD.
-
- notify type indique le type d'événement qui affecte l'onduleur :
 - ONLINE : onduleur alimenté par le secteur.
 - ONBATT : onduleur sur batteries.
 - LOWBATT: la batterie de l'onduleur est faible.
 - SHUTDOWN : arrêt du système.
 - FSD : le maître demande l'arrêt forcé de l'onduleur.
 - REPLBATT : batteries défectueuses.

VI.9. LANCEMENT DE NUT

NUT peut être lancé avec la commande : **>/etc/init.d/nut start**

L'UPS est ainsi configuré et fonctionnel.

Pour s'en assurer, aller voir dans **/var/log/daemon.log** :

Un résultat similaire devrait apparaître :

```
Jan 21 22:39:31 Etch megatec[2661]: Startup successful
Jan 21 22:39:31 Etch upsd[2662]: Connected to UPS [ipel]: megatec-ttyS0
Jan 21 22:39:32 Etch upsd[2663]: Startup successful
Jan 21 22:39:32 Etch upsmon[2665]: Startup successful
Jan 21 22:39:32 Etch upsd[2663]: Connection from 127.0.0.1
Jan 21 22:39:32 Etch upsd[2663]: Client localhost@127.0.0.1 logged into UPS [ipel]
```

La première ligne montre que le driver (megatec) est correctement chargé, l'UPS est bien détecté.

VI.10. LE DELAI AU BOUT DUQUEL LE SERVEUR VA S'ETEINDRE



Par défaut, dans la configuration standard qui a été faite, NUT attend que la batterie soit presque vide (ce qu'il détecte via l'état de l'UPS) pour déclencher l'extinction du système.

Le système exécute alors automatiquement un shutdown -h now et tous les process s'éteignent les uns après les autres jusqu'à l'arrêt complet de la machine. L'UPS a rempli son rôle.

VI.10.1. Le paramétrage du démon upsmon

Pour contrôler ce délai, il faut reprendre le fichier de configuration /etc/nut/upsmon.conf :

```
# définition des UPS à surveiller et des autorisations

MONITOR ipel@localhost 1 admin adminpw master

MINSUPPLIES 1

# définition de la commande Shutdown
SHUTDOWNCMD "/sbin/shutdown -h +0"

# lancer le programme à exécuter le script avec retard

NOTIFYCMD /sbin/upssched
NOTIFYFLAG ONBATT SYSLOG+WALL+EXEC
NOTIFYFLAG ONLINE SYSLOG+WALL+EXEC

POLLFREQ 5
POLLFREQUALERT 5
HOSTSYNC 15
DEADTIME 15
POWERDOWNFLAG /etc/killpower
FINALDELAY 5
```

VI.10.2. Le paramétrage de « Upssched »

"upssched" est un utilitaire ajouté au package de NUT pour gérer l'exécution de scripts en différé.

- Créer le fichier **/etc/nut/upssched.conf**

```
# Le script qui sera exécuté
CMDSCRIPT /etc/nut/upsalert

# les champs obligatoires qui doivent être mis devant les commandes AT
PIPEFN /var/run/upssched/upssched.pipe
LOCKFN /var/run/upssched/upssched.lock

# les minuteries, ici 10 secondes après le événement ONBATT(ups sur batterie)
AT ONBATT * START-TIMER onbatt 10

# annuler le compte à rebours lorsque la mise sous tension revient
AT ONLINE * CANCEL-TIMER onbatt
```

Il y sera indiqué le timing sur l'événement ONBATT.

Précisions :

* : désigne tous les UPS géré par NUT

START-TIMER : pour dire que le compte à rebours commence.

"onbatt" : désigne le nom du timer (car il peut y avoir plusieurs événements à des timings différents, donc il faut un nom pour les différencier)



le chiffre 30 : est le timeout en secondes, c'est-à-dire le délai au bout duquel le script sera exécuté.
CANCEL-TIMER onbatt : commande d'annuler le compte à rebours si l'événement ONLINE survient.

- Créer un fichier **/etc/nut/upsalert**.

Le code qui sera exécuté est le suivant :

```
#!/bin/sh

# écrire dans un fichier log spécifique

date=$(date +"%F %T")
echo "$date >> Panne de courant, l'UPS va éteindre le serveur..." >> /var/log/ups.log

# envoyer le signal d'arrêt au master (sera diffusé à tous les esclaves)
/sbin/upsmon -c fsd
```

- Donner la permission exécutable au script (chmod 755).

A noter : pour gérer plusieurs événements, un seul script est autorisé, il faudra donc un moyen pour détecter le type d'événement à traiter.

Pour cela NUT définit un paramètre qui est envoyé au script (\$*).

- Créer un fichier **/var/log/ups.log** avec en première ligne :

```
# cd /var/log/
# vi ups.log
« Alertes UPS » + retour chariot
```

- Créer un répertoire **/var/run/upssched** dédié à upssched qui a besoin d'écrire temporairement quelques données sur fichier, mais il faut lui spécifier où il peut le faire et régler les permissions appropriées.

```
# cd /var/run/
# mkdir upsshed
```

Ce dossier sera légué au groupe "nut" (chown nut :nut) et nut aura l'accès en écriture (chmod 775)

VI.11. TESTER LA CONFIGURATION

Pour tester si nut est bien configuré, il faut débrancher l'onduleur. Il s'en suit un shutdown immédiat qui correspond à la dernière instruction du script upsalert.

Au redémarrage de la machine, nous pouvons constater qu'il y a une trace dans **/var/log/ups.log**.

Il est également possible d'utiliser la commande: **upsc UPS@localhost** pour voir si l'onduleur est bien configuré. Cette commande affiche toutes les informations disponibles de l'onduleur.

Résultat de la commande upsc ipel@localhost:

```
Etch:/etc/init.d# upsc ipel@localhost
battery.charge: 97.5
battery.voltage: 13.60
battery.voltage.nominal: 12.0
driver.name: megatec
driver.parameter.port: /dev/ttyS0
driver.version: 2.0.4
driver.version.internal: 1.4
input.frequency: 50.0
input.voltage: 230.4
```



```
input.voltage.fault: 230.4
input.voltage.maximum: 232.7
input.voltage.minimum: 226.7
output.voltage: 227.2
output.voltage.nominal: 220.0
ups.delay.shutdown: 2
ups.delay.start: 3
ups.load: 7.0
ups.mfr: unknown
ups.model: unknown
ups.serial: unknown
ups.status: OL
ups.temperature: 25.0
```

Pour n'avoir qu'une seule information, nous pouvons ajouter à la commande un argument.
Par exemple **upsc ipel@localhost battery.charge**

```
Etch:/etc/init.d# upsc ipel@localhost battery.charge
97.5
```

VI.12. LA PROGRAMMATION DE L'UPS

Suivant le matériel utilisé, il est envisageable de programmer l'onduleur. Dans ce cas d'étude, l'Infosec IPEL ou l'APC le permettent.

Il est possible, par exemple, de signifier à l'onduleur qu'il doit attendre 2 min après la demande de shutdown avant de couper effectivement.

lancer la commande **upsrw UPS@localhost** pour voir les programmations possibles

résultat :

```
Etch:/etc/init.d# upsrw ipel@localhost
[ups.delay.shutdown]
Interval to wait after shutdown with delay command (seconds)
Type: STRING
Value: 2

[ups.delay.start]
Interval to wait before (re)starting the load (seconds)
Type: STRING
Value: 3
```

ups.delay.shutdown : l'UPS attend 0 min après la demande de shutdown avant de couper effectivement
ups.delay.start : l'UPS attend 3 min après le retour du courant avant de démarrer

De la même façon, pour d'autres onduleurs, il sera possible de configurer, par exemple, les paramètres suivant :

ups.id : identifiant ****interne**** de l'onduleur, au cas où il n'aurait pas été étiqueté

battery.runtime.low : point critique quand il ne reste que 2 min de fonctionnement sur batterie

battery.charge.restart : pour attendre que la batterie soit rechargée au minimum à 15% avant de relancer

ups.delay.shutdown : pour attendre un certain délai après la demande de shutdown avant de couper effectivement

ups.delay.start : pour attendre un délai après le retour du courant avant de démarrer

ups.test.interval : pour effectuer des auto-tests de l'onduleur toutes les semaines

exemple de commande : **# upsrw -s ipel.delay.shutdown=2 -u admin -p admin ipel@localhost**
Cette commande modifie le délai d'attente avant le shutdown.

résultat pour l'Ipel après rappel de la commande **upsrw ipel@localhost**:



```
Etch:/etc/init.d# upsrw ipel@localhost
[ups.delay.shutdown]
Interval to wait after shutdown with delay command (seconds)
Type: STRING
Value: 0

[ups.delay.start]
Interval to wait before (re)starting the load (seconds)
Type: STRING
Value: 3
```

Le délai d'attente avant le shutdown a bien été modifié.

VII. Configuration de NUT en mode Esclave

La configuration côté slave est beaucoup plus simple.

VII.1. INSTALLATION DE NUT

Installer le package nut par la commande :

```
/etc/init.d/# sh install nut
```

VII.2. CONFIGURATION DU FICHIER **/ETC/DEFAULT/NUT**

Dans **/etc/default/nut** :

```
# démarrage du demon upsd
START_UPSD=no

# démarrage du demon upsmon
START_UPSMON=yes
```

Seul le démon upsmon est démarré.
upsd n'est nécessaire que sur la machine branchée physiquement à l'UPS (c'est-à-dire le master).

VII.3. CONFIGURATION DU DEMON UPSMON

Dans un premier temps, installer le paquet nut par la commande :

```
# apt-get install nut
```

Un unique fichier de configuration est nécessaire : **/etc/nut/upsmon.conf**

```
# définition des UPS à surveiller et des autorisations

MONITOR ipel@localhost 1 admin adminpw slave

# définition de la commande Shutdown
SHUTDOWNCMD "/sbin/shutdown -h +0"
```

Dans ce fichier, sont indiqués : l'adresse IP du serveur et le mot "slave" à la place de "master".
Le reste est identique au fichier de configuration vu précédemment dans le cas du master.

Enfin, ne surtout pas oublier d'attribuer des droits à nut. nous suivrons la même procédure que pour le serveur Master

Il faut démarrer NUT par **/etc/init.d/sh nut start**



Pour tester si le serveur esclave est bien configuré, nous simulerons une panne de courant.

Les 2 ordinateurs s'éteindront maintenant automatiquement, le serveur ayant donné le signal à tous les esclaves qui lui sont connectés.

VIII. Configuration de NUT sur Windows (Esclave)

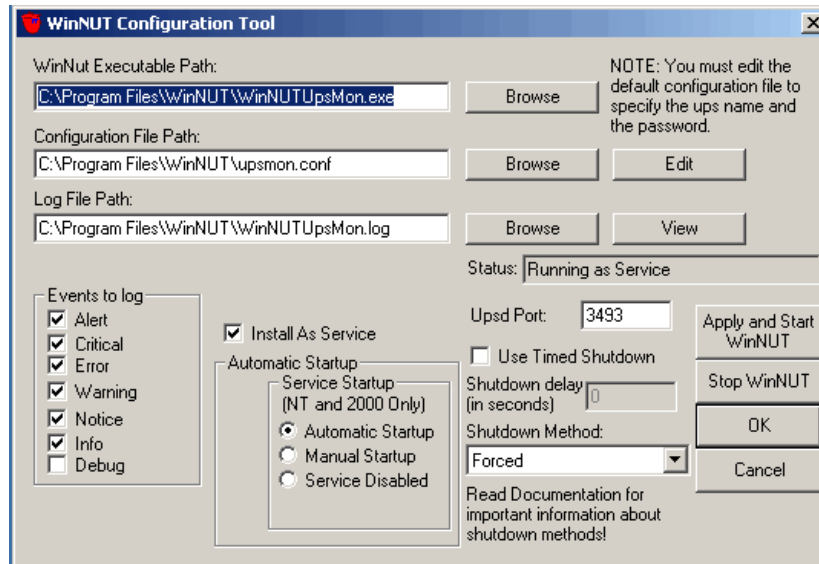
Il est impératif d'utiliser le portage de NUT pour Windows afin pour pouvoir communiquer avec le serveur NUT maître sous linux.

La version de NUT pour Windows ne reprend pas 100% de l'application, mais uniquement la partie cliente, c'est-à-dire le monitoring réseau.

Une machine Windows ne pourra donc jamais être master avec NUT, ce qui dans notre cas n'est pas un problème puisque nous avons un serveur linux jouant déjà ce rôle.

VIII.1. INSTALLATION DE WINNUT

Démarrer le programme de configuration dans le menu **Démarrer - Programmes - WinNUT - WinNUT Configuration Tool** :



VIII.2. CONFIGURATION DE WINNUT

Les options à régler :

- cocher "**Install As Service**"
- cocher "**Automatic Startup**", ces 2 options assurant que WinNUT sera exécuté automatiquement à chaque démarrage de Windows, en tant que service système
- vérifier que "**Use Timed Shutdown**" est décoché
- « **Shutdown Method** » : forced

VIII.3. PARAMETRAGE DU FICHIER UPSMON

Paramétrer le fichier de configuration **upsmon.conf**, comme sous linux.

Cliquer sur le lien "Edit" en face du nom du fichier



Le fichier est très complet, toutes les options sont déjà réglées à leur valeur par défaut et le fichier est bien commenté ! Ici seule l'option "MONITOR" devra être réglée : **MONITOR ipel@localhost 1 admin adminpw slave**

Enregistrer le fichier et quitter l'éditeur.

De retour dans l'outil de configuration de WinNUT, cliquer sur le bouton "**Apply and start WinNUT**".

La machine Windows est opérationnelle et surveille l'état de l'UPS grâce au démon linux.

IX. Les outils de surveillance

Ils permettront de surveiller périodiquement l'état d'UPS gérés par un démon NUT afin d'en faire un affichage graphique.

Interface graphique, WinNUTClient, PRTG

IX.1. CONFIGURATION DU DEMON SNMP

IX.1.1. Principe de fonctionnement du protocole SNMP

Le système de gestion de réseau est basé sur deux éléments principaux: un superviseur et des agents.

Le superviseur est la console qui permet à l'administrateur réseau d'exécuter des requêtes de management. Les agents sont des entités qui se trouvent au niveau de chaque interface connectant l'équipement managé au réseau et permettant de récupérer des informations sur différents objets.

Tout équipement contient des objets manageables qui peuvent être des informations matérielles, des paramètres de configuration, des statistiques de performance et autres objets qui sont directement liés au comportement en cours de l'équipement en question.

Ces objets sont classés dans une sorte de base de donnée appelée MIB ("Management Information Base").

SNMP permet le dialogue entre le superviseur et les agents afin de recueillir les objets souhaités dans la MIB.

L'architecture de gestion du réseau proposée par le protocole SNMP est donc basée sur trois principaux éléments :

- Les équipements managés (managed devices) sont des éléments du réseau (ponts, hubs, routeurs ou serveurs), contenant des "objets de gestion" (managed objects) pouvant être des informations sur le matériel, des éléments de configuration ou des informations statistiques ;
- Les agents, c'est-à-dire une application de gestion de réseau résidant dans un périphérique et chargé de transmettre les données locales de gestion du périphérique au format SNMP ;
- Les systèmes de management de réseau (network management systems notés NMS), c'est-à-dire une console au travers de laquelle les administrateurs peuvent réaliser des tâches d'administration.

Pour récupérer les informations de fonctionnement de l'onduleur afin de pouvoir les exploiter sur une autre machine, il est nécessaire d'installer et de configurer SNMP.

IX.1.2. Installation

Il faut installer le paquet **snmpd**

Cette installation se fait par la commande :

```
# apt-get install snmpd
```

IX.1.3. Configuration

Le fichier de configuration se situe sous **/etc/snmp/snmpd.conf**

Remplacer toutes les lignes par celles-ci, à adapter aux besoins si nécessaire :



```
syscontact nath@lyceeSacreCoeur.net
syslocation Le placard
```

```
# 1° créer des relations entre les communautés et des noms de sécurité
# nom.secu source communaute
```

```
com2sec Local 127.0.0.1 private
com2sec LocalNet 192.168.40.0/24 public
```

```
# 2° créer des relations entre des noms de groupes et les noms de sécurité
# nom.groupe version nom.secu
```

```
group RWGroup v1 Local
group ROGroup v1 LocalNet
```

```
#3° Créer les diverses vues qui seront autorisées aux groupes
#
```

```
view tout included .1 80
```

```
#4° Indiquee les accès aux vues suivant les groupes
```

```
# nom.groupe contexte modele.secu niveau.secu prefixe lecture ecriture notification
```

```
access ROGroup "" v1 noauth exact tout none none
access RWGroup "" v1 noauth exact tout tout none
```

```
#Points de montage à surveiller
```

```
disk / 100000
disk /home 100000
disk /var 100000
disk /lib/init/rw 100000
disk /dev/shm 100000
```

```
# Interface réseau
```

```
interface eth0 6 10000000
```

IX.1.4. Redémarrage du demon snmpd

Redémarrer le demon snmpd avec la commande :

```
/etc/init.d/# sh snmpd restart
```

IX. 2. L'INTERFACE WEB

Afin de pouvoir contrôler les paramètres de l'onduleur sur le Web, il existe un outil nut-cgi qui permet de diffuser ces informations. Il est disponible dans le paquet NUT.

1. installer `nut-cgi` avec la commande

```
#apt-get install nut-cgi
```

2. copier les modèles de documents disponibles avec la commande :

```
#cp -a /usr/share/doc/nut-cgi/examples/* /etc/nut/
```



3. modifier le fichier /etc/nut/hosts.conf :

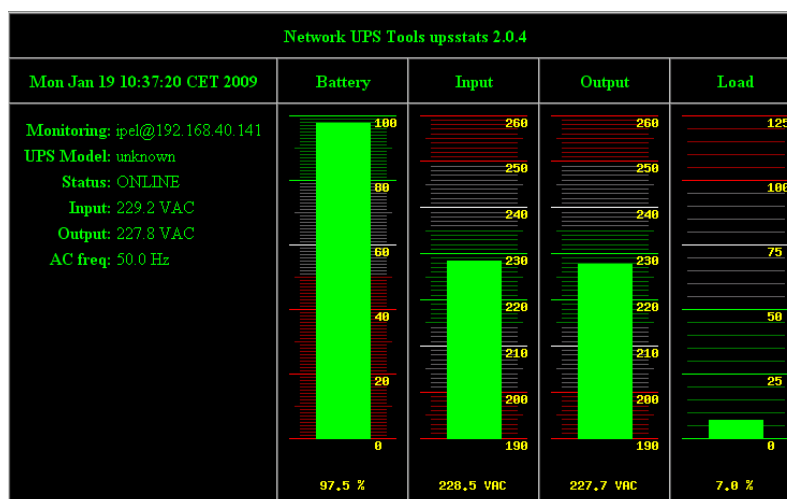
```
# /etc/nut/hosts.conf
MONITOR ipel@192.168.40.141 "Ipel"
```

4. consulter la page <http://192.168.40.141/cgi-bin/nut/upsstats.cgi>

un résultat suivant apparaît :

Network UPS Tools upsstats 2.0.4							
Mon Jan 19 10:37:16 CET 2009							
System	Model	Status	Battery	Input	Output	Load	UPS Temp
Ipel	unknown	ONLINE	97.5 %	229.3 VAC	227.7 VAC	7.0 %	25.0 °C

En cliquant sur [Ipel](#) (à gauche), l'écran suivant apparaît affichant les informations de l'onduleur.

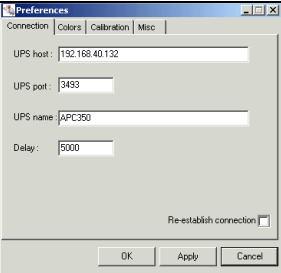
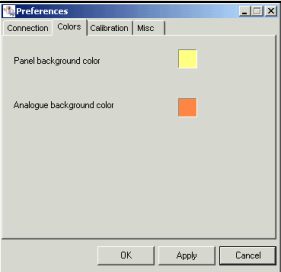
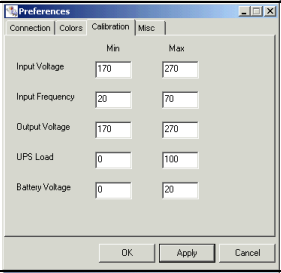
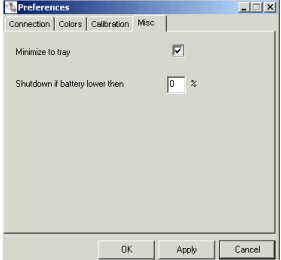


Les paramètres comme la couleur de fond, la police de caractères, par exemple, sont configurables dans le fichier /etc/nut/upsstats.html. Cela peut rendre l'interface Web plus conviviale.

IX.3. CONFIGURATION DE WINNUTCLIENT

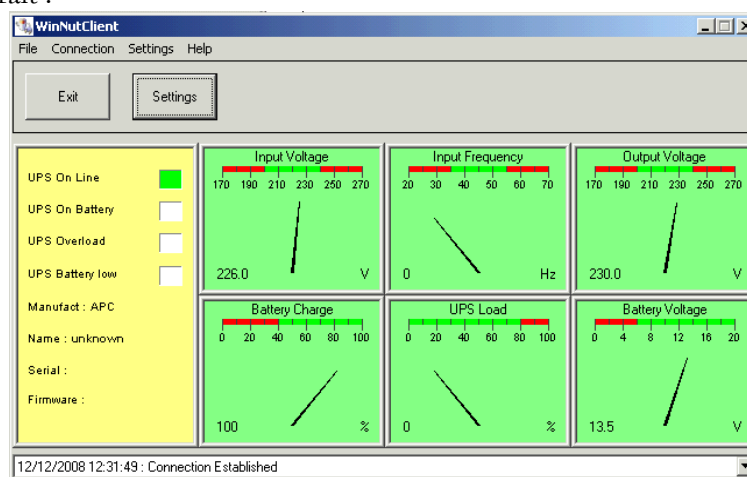
WinNUTClient est une interface graphique permettant la surveillance des paramètres de l'onduleur.

1. Télécharger WinNutClient Version 1.5
2. Extraire les fichiers du zip
3. Exécuter le fichier **.exe**
4. Cliquer le Bouton Setting Préférences :

	<p>Dans l'onglet Connection, renseigner les champs suivants :</p> <p>UPS Host : Adresse IP du Serveur</p> <p>UPS Name : Nom de l'UPS</p>
	<p>L'onglet Color permet de sélectionner les couleurs de fond de l'écran de contrôle</p>
	<p>L'onglet Calibration permet de paramétrer les appareils de mesure en fonction des données constructeur de l'onduleur</p>
	<p>Dans l'onglet MISC :</p> <p>Sélectionner Minimize to Tray pour avoir la possibilité de réduire la fenêtre.</p>

A noter : Toutes les informations entrées dans ces différents onglets sont accessibles dans le fichier *ups.txt*

Le résultat suivant apparait :



IX.4. PRTG

PRTG est un logiciel de supervision de réseau.



Pour voir l'onduleur dans PRTG, il faut tout d'abord importer la Mib du constructeur. Dans ma présente étude, il s'agit d'Infossec. Après avoir cherché très longtemps, il ne m'a pas été possible de la récupérer. Par contre, pour les onduleurs APC, leur Mib est disponible.

Les tests n'ont pas pu être effectués mais le principe est d'installer Paessler MIB Importer. Il s'agit d'un logiciel qui permet d'importer des Mibs dans PRTG.

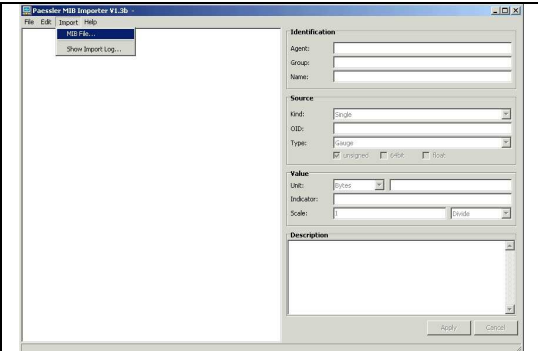
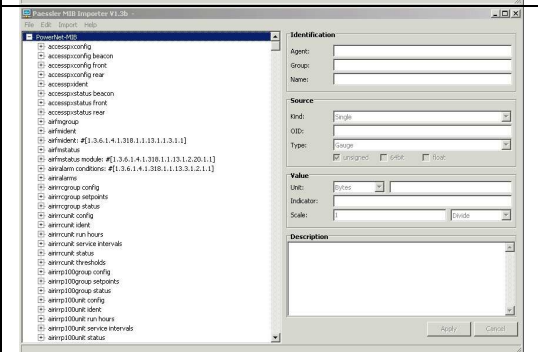
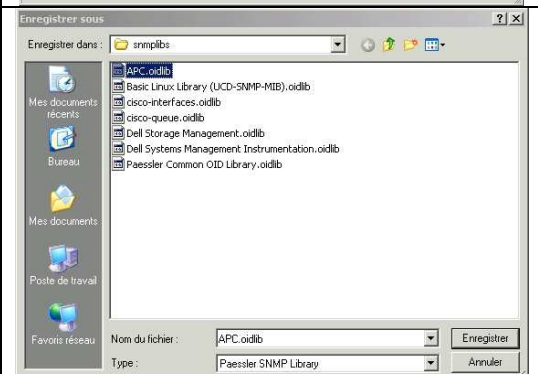
Une fois la Mib montée dans la librairie PRTG, il reste à trouver les bonnes sondes pour remonter les informations de l'UPS dans PRTG et ainsi d'établir une surveillance au niveau du poste de supervision.

Le principe :

Pour que la Mib soit reconnu dans PRTG, il faut :

Télécharger le fichier Mib APC à l'adresse : <http://support.ipmonitor.com/mibs/POWERNET-MIB/info.aspx>

Installer le logiciel Paessler MIB Importer

	<p>Dans l'onglet « import », sélectionner le fichier Mib téléchargé et valider en sélectionnant « ouvrir »</p>
	<p>Une fois la Mib importée, sélectionner, dans l'onglet <file> « Save for.PRTG Traffic Grapher ».</p>
	<p>La Mib sera automatiquement montée dans la bibliothèque de PRTG après lui avoir donné un nom..</p>

Pour les raisons que j'évoquais ci-dessus, je n'ai pas pu faire de tests probants dans PRTG.

