

La gestion de parc informatique.

Présentation :

Le parc informatique d'une organisation est un assemblage de matériels et de logiciels accumulés tout au long des années. On y trouve des matériels différents (téléphones, portables, ordinateurs fixes etc.) qui peuvent être de plusieurs générations, logiciels et OS variés (Linux, Windows, Mac OS).

Nous allons utiliser l'application OCS NG qui permet notamment II permet notamment d'automatiser les inventaires des ordinateurs connectés sur le réseau et de leurs composants matériels et logiciels, de connaître l'ensemble des équipements du parc informatique (matériels et logiciels) et de téléistribuer des fichiers et des applications.

Objectifs :

A la fin de ce tutoriel, vous devez être capable :

- D'installer l'application OCSInventory.
- D'installer un agent sur les machines clients sur deux OS différents.
- D'installer et configurer un serveur HTTPS.
- De déployer une application.

Pré requis :

- ➔ Ordinateur sur Linux (ici, on utilisera la Debian 7.4).
- ➔ Avoir une connexion internet.
- ➔ Avoir une IP fixe pour le serveur OCS.

Sommaire :

- I. Installation d'un Service Web et d'un SGBD
- II. Installation et configuration du Service OCS
- III. La collecte d'information
- IV. Configuration d'un serveur HTTPS
- V. Le déploiement d'application

I. Installation d'un Service WEB et d'un SGBD.

Ping avec la passerelle par défaut de la box *192.168.1.254* pour savoir si nous pouvons sortir du domaine afin de pouvoir télécharger les paquets nécessaire :

```
root@ServeurOCSPM:~# ping 192.168.1.254
PING 192.168.1.254 (192.168.1.254) 56(84) bytes of data.
64 bytes from 192.168.1.254: icmp_req=1 ttl=64 time=1.44 ms
64 bytes from 192.168.1.254: icmp_req=2 ttl=64 time=0.956 ms
64 bytes from 192.168.1.254: icmp_req=3 ttl=64 time=0.733 ms
64 bytes from 192.168.1.254: icmp_req=4 ttl=64 time=0.792 ms
64 bytes from 192.168.1.254: icmp_req=5 ttl=64 time=0.648 ms
64 bytes from 192.168.1.254: icmp_req=6 ttl=64 time=0.843 ms
^C
--- 192.168.1.254 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5009ms
rtt min/avg/max/mdev = 0.648/0.903/1.446/0.260 ms
```

Configuration du fichier */etc/network/interfaces* pour se mettre en IP fixe :

```
#Configuration de l'adresse en fixe
auto eth0
iface eth0 inet static
address 192.168.1.222
netmask 255.255.255.0
gateway 192.168.1.254
```

Puis, nous vérifions l'adresse IP :

```
root@ServeurOCSPM:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 86:7b:41:69:48:30
          inet adr:192.168.1.222  Bcast:192.168.1.255  Masque:255.255.255.0
```

Mise à jour des paquets avant chaque installation:

```
root@ServeurOCSPM:~# apt-get update
```

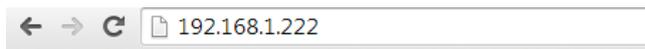
Installation du Service WEB pour le https qui est nécessaire à OCS :

```
root@ServeurOCSPM:~# apt-get install apache2
```

Apache2 est bien installé et s'exécute :

```
root@ServeurOCSPM:~# service apache2 status
Apache2 is running (pid 3020).
```

Puis dans un navigateur, nous accédons bien à l'index.html :



It works!

This is the default web page for this server.

The web server software is running but no content has been added, yet.

Service WEB du Serveur OCS de Pierre-Marie

Installation du SGBD (MySQL) pour pouvoir stocker des informations dans une base de données :

```
root@ServeurOCSPM:~# apt-get install mysql-server mysql-client
```

II. Installation et configuration du service OCSInventory :

a) Installation OCSInventory

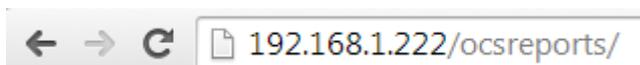
Mise à jour des paquets :

```
root@ServeurOCSPM:~# apt-get update
```

Installation du paquet OCSInventory :

```
root@ServeurOCSPM:~# apt-get install ocsinventory-server ocsinventory-reports
```

Plus qu'à finir l'installation avec l'interface WEB avec :



Puis remplir comme suit (MySQL password est le même mot de passe que celui tapé lors de l'installation de MySQL) :

MySQL login:	<input type="text" value="root"/>
MySQL password:	<input type="password" value="...."/>
Name of Database:	<input type="text" value="ocsweb"/>
MySQL HostName:	<input type="text" value="localhost"/>
<input type="button" value="Send"/>	

Et nous rafraichissons la page web pour arriver à cette interface (le password est admin par défaut) :

User:

Password:

Enfin, nous activons les logs :

1

2

3; on met sur "on" pour activer les logs

LOGLEVEL
Fonctionnalité de log sur le serveur

ON
 OFF

PROLOG_FREQ
Fréquence de lancement de l'agent par le service

24 heures
(Doit être supérieur ou égal à 1)

III. La collecte d'information.

La collecte automatisée d'informations passe par l'installation sur les postes clients par un agent OCS. Nous allons voir maintenant l'installation sur deux systèmes d'exploitation différents.

a) Installation de l'agent sous Linux Debian :

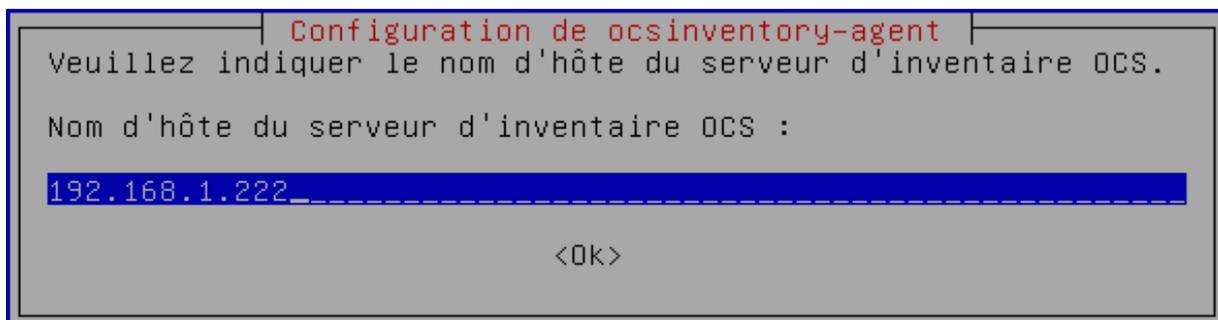
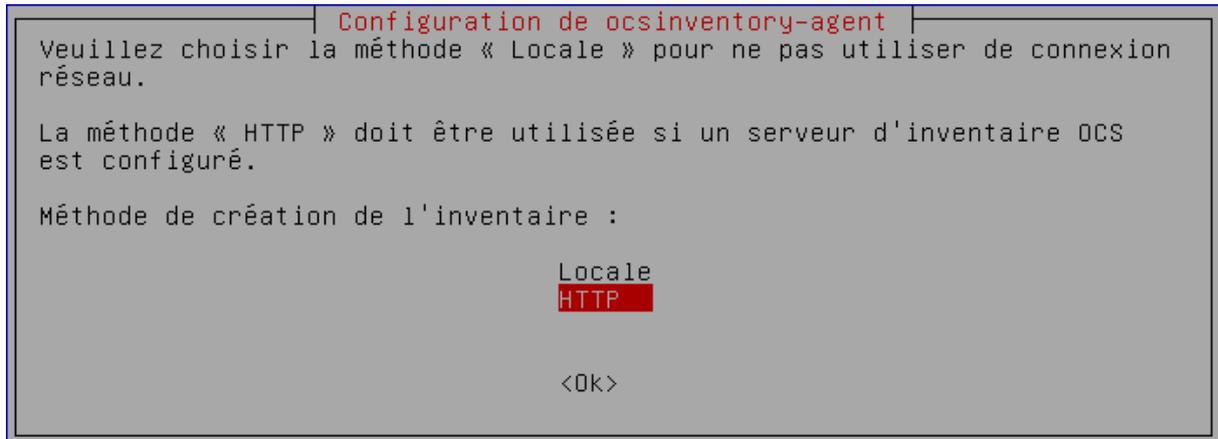
Mise à jour des paquets :

```
root@ServeurOCSPM:~# apt-get update
```

Installation du paquet de l'agent :

```
root@ServeurOCSPM:~# apt-get install ocsinventory-agent
```

Nous choisissons la méthode http qui permet de remonter les informations à un serveur OCS :



Enfin, pour forcer le déclenchement du cron la première fois, nous allons taper la commande *ocsinventory-agent*

```
root@ServeurOCSPM:~# ocsinventory-agent
[info] Accountinfo file doesn't exist. I create an empty one.
[info] PROLOG_FREQ has changed since last process(old=,new=24)
[info] No support registered for your installation. Check OCS Inventory NG support packages at http://www.ocsinventory-ng.com
[info] [download] Download is off.
[info] [download] Beginning work. I am 13706.
[info] [download] Option turned off. Exiting.
```

Puis, nous allons aller sur l'interface WEB, nous cliquons sur l'onglet encadré en rouge :



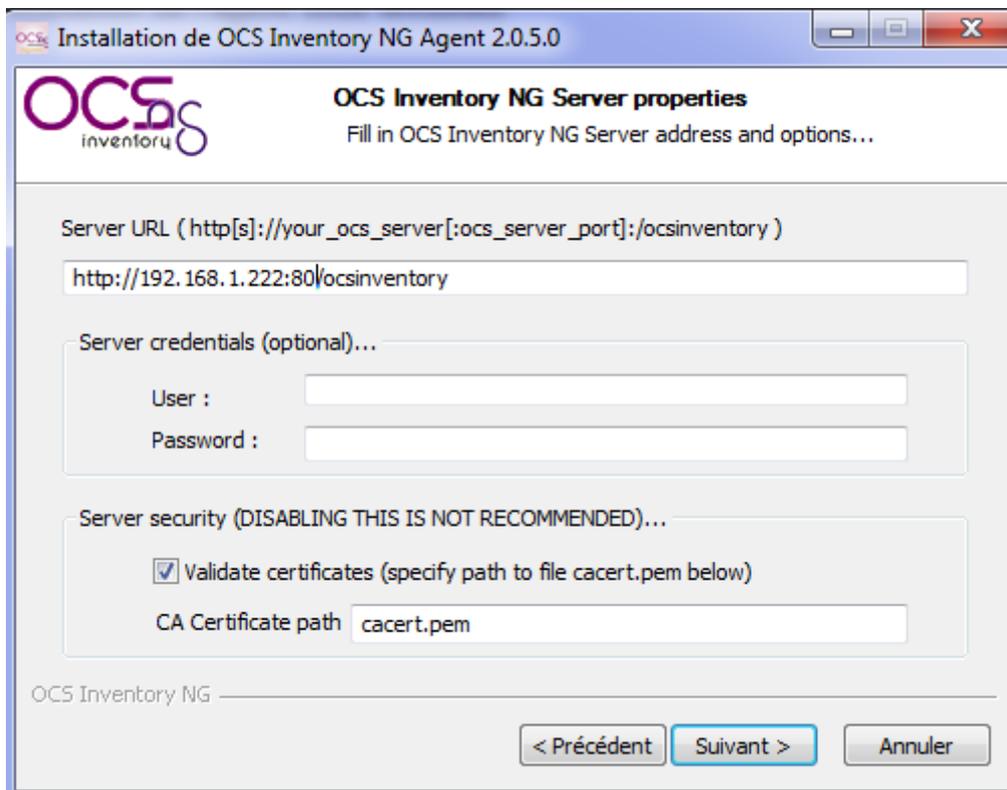
Et nous distinguons le Serveur OCS. Si nous cliquons sur la machine, nous avons plus d'information :

Account info: TAG	▲ Dernier inventaire	Machine	Utilisateur	Systeme	RAM(MB)	CPU(MHz)	Sélectionner	Supprimer
NA	2015-04-08 16:01:22	ServeurOCSPM	root	Debian GNU/Linux 7.8 (wheezy)	496	2133	<input type="checkbox"/>	<input checked="" type="checkbox"/>

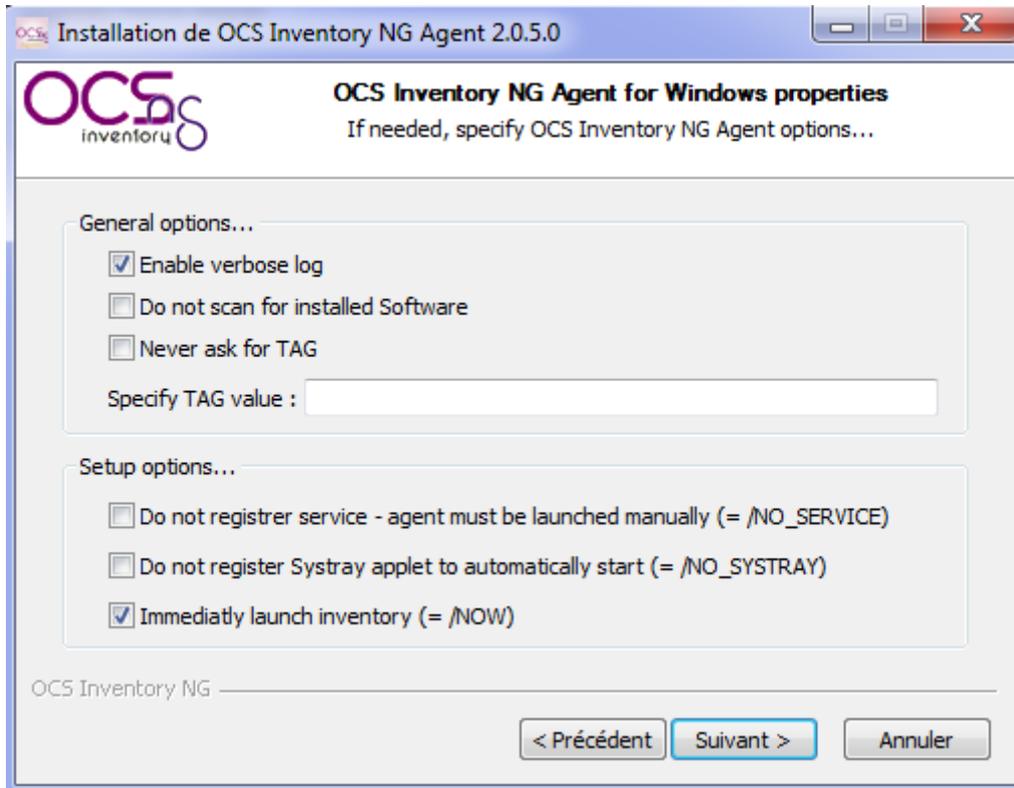
Nom:	ServeurOCSPM	Domaine:	sio.local
Adresse IP:	192.168.1.222	Utilisateur:	root
Espace de Swap:	461	Nom du système:	Debian GNU/Linux 7.8 (wheezy)
Version du système:	3.2.0-4-amd64	Service pack:	#1 SMP Debian 3.2.65-1
Type agent:	OCS-NG_unified_unix_agent_v2.0.5	Mémoire:	512
Dernier inventaire:	08/04/2015 16:01:22	Dernier contact:	08/04/2015 16:01:22
Description:	x86_64/00-00-00 01:03:26	Uuid:	# SMBIOS implementations newer than version 2.7 are not # fully supported by this version of dmidecode. 70947A6D-F4FD-834B-AC7A-712F7A5D7664
Exporter:	XML		

b) Installation de l'agent sous Windows

Lancer l'exécutable Windows-Agent-Setup et remplir l'url correspondant (le :80 est le port http) :



Mettre un proxy s'il y en a un puis cocher les cases correspondantes :



Maintenant, nous vérifions la remontée dans l'interface WEB :

Account info: TAG	△ Dernier inventaire	Machine	Utilisateur	Système	Adresse IP	Sélectionner	Supprimer
Salle 105	2015-04-08 16:41:06	POSTE05	etudiantsio	Microsoft Windows 7 Professionnel	192.168.1.55	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Salle 105	2015-04-08 16:28:09	POSTE06	etudiantsio	Microsoft Windows 7 Professionnel	192.168.1.56	<input type="checkbox"/>	<input checked="" type="checkbox"/>
VM Noeud 5 105	2015-04-08 16:01:22	ServeurOCSPM	root	Debian GNU/Linux 7.8 (wheezy)	192.168.1.222	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Nom:	POSTE06	Domaine:	sio.local
Adresse IP:	192.168.1.56	Utilisateur:	etudiantsio
Espace de Swap:	8074	Nom du système:	Microsoft Windows 7 Professionnel
Version du système:	6.1.7601	Service pack:	Service Pack 1
Utilisateur Windows:	etudiantsio	Licence Windows:	55041-001-4875997-86200
Clé Windows:	BBBBB-BBBBBB-BBBBBB-BBBBBB	Type agent:	OCS-NG_WINDOWS_AGENT_v2.0.5.0
Mémoire:	8192	Dernier inventaire:	08/04/2015 16:28:09
Dernier contact:	08/04/2015 16:28:09	Description:	poste
Uuid:	03D40274-0435-05E2-1606-9E0700080009	Exporter:	XML

Nom:	POSTE05	Domaine:	sio.local
Adresse IP:	192.168.1.55	Utilisateur:	etudiantsio
Espace de Swap:	9098	Nom du système:	Microsoft Windows 7 Professionnel
Version du système:	6.1.7601	Service pack:	Service Pack 1
Utilisateur Windows:	etudiantsio	Licence Windows:	55041-001-4875997-86200
Clé Windows:	BBBBB-BBBBBB-BBBBBB-BBBBBB	Type agent:	OCS-NG_WINDOWS_AGENT_v2.0.5.0
Mémoire:	8192	Dernier inventaire:	08/04/2015 16:41:06
Dernier contact:	08/04/2015 16:41:06	Uuid:	03D40274-0435-05E2-1206-350700080009
Exporter:	XML		

IV. Configuration d'un serveur HTTPS.

Créer un certificat pour le serveur web (*nano apache_generate_cert.sh*) puis :

```
GNU nano 2.2.6      Fichier : apache_generate_cert.sh
openssl genrsa -out server.key 1024
openssl req -outform PEM -new -key server.key -x509 -days 1825 -out server.crt
```

Nous changeons les droits d'exécution du script (U pour user et X pour exécution):

```
root@ServeurOCSPM:/# chmod u+x apache_generate_cert.sh
```

Puis nous lançons le script :

```
root@ServeurOCSPM:/# sh apache_generate_cert.sh
```

Puis remplir les informations suivantes :

```
Country Name (2 letter code) [AU]:FR
State or Province Name (full name) [Some-State]:Bn
Locality Name (eg, city) []:Caen
Organization Name (eg, company) [Internet Widgits Pty Ltd]:SIO
Organizational Unit Name (eg, section) []:BTS
Common Name (e.g. server FQDN or YOUR name) []:192.168.1.222
Email Address []:
```

Enfin, nous chargeons le module :

```
root@ServeurOCSPM:/# a2enmod ssl
```

Nous copions les fichiers du certificat :

```
root@ServeurOCSPM:/# cp /server.* /etc/ssl/private
```

Nous allons mettre à jour les fichiers de configuration d'apache2 :

```
root@ServeurOCSPM:/# nano /etc/apache2/sites-available/default-ssl
```

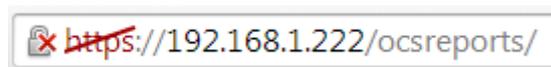
Et modifier par ceci :

```
SSLCertificateFile /etc/ssl/private/server.crt
SSLCertificateKeyFile /etc/ssl/private/server.key
```

Puis taper ces commandes :

```
root@ServeurDCSPM:/# a2ensite default-ssl
root@ServeurDCSPM:/# service apache2 restart
```

Maintenant, nous pouvons nous connecter avec l'https :



Enfin, il faut copier le certificat server.crt et le copier dans le fichier osc sur les clients ocs windows.

V. Le déploiement d'une application.

The screenshot shows the configuration page for 'Téléchargement' (Download) in the OCS Inventory interface. The 'Téléchargement' tab is selected and highlighted with a red box. Below the tabs, several configuration options are listed, each with a description and a control element. Several of these options are highlighted with red boxes:

- DOWNLOAD**: Fonctionnalité de télédéploiement (agent et serveur). ON, OFF.
- DOWNLOAD_CYCLE_LATENCY**: Temps d'attente entre 2 cycles de télédéploiement. Value: 60 secondes. (Doit être supérieur ou égal à 1)
- DOWNLOAD_FRAG_LATENCY**: Temps d'attente entre 2 fragments téléchargés. Value: 10 secondes. (Doit être supérieur ou égal à 1)
- DOWNLOAD_GROUPS_TRACE_EVENTS**: Spécifie si vous souhaitez suivre les paquets affectés à un groupe de niveau ordinateur. ON, OFF.
- DOWNLOAD_PERIOD_LATENCY**: Temps d'attente entre 2 périodes de télédéploiement. Value: 15 secondes. (Doit être supérieur ou égal à 1)
- DOWNLOAD_TIMEOUT**: Validité d'un paquet à compter de sa prise en compte par l'agent. Value: 30 jours. (Doit être supérieur ou égal à 1)
- DOWNLOAD_PERIOD_LENGTH**: Priorité maximum des paquets téléchargés (Les paquets d'une priorité supérieure sont ignorés). Value: 10
- DEPLOY**: Fonctionnalité de déploiement automatique de l'agent. ON, OFF.
- DOWNLOAD_URI_FRAG**: Adresse où se trouvent les fragments des paquets de télédéploiement à activer. Par défaut (HTTP://localhost/download), Personnaliser. Value: http://192.168.1.222/download
- DOWNLOAD_URI_INFO**: Adresse où se trouvent les fichiers INFO des paquets de télédéploiement à activer. Par défaut (HTTPS://localhost/download), Personnaliser. Value: https://192.168.1.222/download



Nous créons le paquet de télé déploiement avec l'icône . Au préalable, il faut créer le répertoire zip sur le client.

Maintenant, nous activons le paquet manuellement puis nous validons :

Activation :

Activation de paquets => Utilitaire_putty (1430137619)

Serveur de fichiers /1430137619

Serveur https /1430137619

La page à l'adresse 192.168.1.222 indique :

Paquet activé, il peut maintenant être affecté

Ensuite, nous affectons le paquet avec recherche multicritère et nous cochons la machine que l'on veut. Enfin nous sélectionnons le paquet avec le SELECT.

Paquets sur les machines | Paquets sur les groupes de redistribution

Afficher:

Restreindre l'affichage:

Ajouter colonne:

2 Résultat(s) (Télécharger)

Nom du paquet	Priorité <input checked="" type="checkbox"/>	Nom du système <input checked="" type="checkbox"/>	TAILLE (KB) <input checked="" type="checkbox"/>	PACK_LOC	SELECT
firefox	5	WINDOWS	40973109	192.168.1.222/download	<input checked="" type="checkbox"/>
Utilitaire_putty	5	WINDOWS	22	192.168.1.222/download	<input checked="" type="checkbox"/>


Action sur le résultat de la requête


1 machine(s) affectée(s) avec succès.

En résultat, on voit qu'une machine a été notifiée :

Timestamp	Date de création	Nom	Notifié	Succès	Erreur	Archives	Activer	Supprimer	Sélectionner	Stats
1430137619	2015-04-27 14:26:59	Utilitaire_putty	1						<input type="checkbox"/>	

Dans le OCSInventory.log, tout c'est bien passé :

```
DOWNLOAD => Download and setup tool successfully started
AGENT => Unloading communication provider
AGENT => Unloading plug-in(s)
AGENT => Execution duration: 00:00:36.
```

Créer un paquet en exécuter en mode silencieux (-ms) pour Firefox:

Création d'un nouveau paquet

Nom:	Firefox
Description:	<u>Deploiement de l'exécutable de l'installation de Firefox</u>
Système:	WINDOWS ▾
Protocole:	HTTP ▾
Priorité:	5 ▾
Fichier (déployé sur les ordinateurs clients):	Choisissez un fichier Firefox.zip
Action:	Exécuter ▾ Commande: "Firefox.exe" -ms
Serveurs de redistribution	
Utilisation sur ce paquet de la redistribution:	NON ▾
Interactions utilisateur	
Prévenir utilisateur:	NON ▾
La fin de l'installation nécessite une intervention utilisateur:	NON ▾

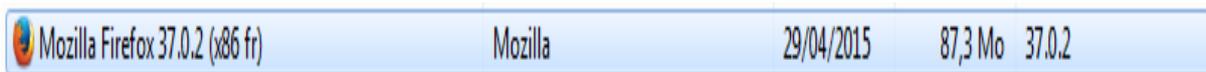
Envoyer

Le téléchargement s'est bien fait vue le log :

```

DOWNLOAD => Executing action <EXECUTE> for package <1430310498>
PACKAGE => Executing command <"Firefox.exe" -ms> for package <1430310498>
PACKAGE => Package <1430310498> successfully executed. Command exit code is <0>
DOWNLOAD => Sending result code <SUCCESS> for package <1430310498>
COM SERVER => Initializing CURL library for sendRequest
COM SERVER => Using CURL without server authentication
COM SERVER => Disabling CURL proxy support
COM SERVER => Enabling CURL SSL server validation support using CA Bundle <cacert.pem>
COM SERVER => Sending HTTP Post request to URL <http://192.168.1.222:80/ocsinventory>
COM SERVER => HTTP Post response received <HTTP Status Code #200>
COM SERVER => Cleaning CURL library
DOWNLOAD => Result code request successfully sent
DOWNLOAD => Pausing for fragment latency (10 seconds)
DOWNLOAD => Pausing for cycle latency (60 seconds)
DOWNLOAD => Pausing for period latency (15 seconds)
DOWNLOAD => Starting new period of 10 cycles
DOWNLOAD => Flushing package queue
DOWNLOAD => Parsing directory <C:\ProgramData\OCS Inventory NG\Agent\download> for packages
DOWNLOAD => No package found, exiting
DOWNLOAD => unloading communication provider
DOWNLOAD => Execution duration: 00:11:09.

```



Paquet de python.msi en mode silencieux avec /qn :

Télécharger le msi puis le zipper.

Choisir exécuter puis ajouter à la commande /qn

```

DOWNLOAD => Processing packages for cycle 10 on wednesday, April 29, 2015 16:21:07
DOWNLOAD => Building package <1430316306>
PACKAGE => Verifying fragment files of package <1430316306>
PACKAGE => Checking free disk space for package <1430316306>
PACKAGE => Building ZIP for package <1430316306>
PACKAGE => Verifying ZIP signature for package <1430316306>
DOWNLOAD => Package <1430316306> built successfully
DOWNLOAD => Executing action <EXECUTE> for package <1430316306>
PACKAGE => Executing command <python.msi /qn> for package <1430316306>

```

Désinstallation du paquet de python.msi avec msiexec /x python.msi /qn :

Télécharger le msi puis le zipper

Choisir exécuter puis ajouter à la commande msiexec /x python.msi /qn

```

DOWNLOAD => Building package <1430312533>
PACKAGE => Verifying fragment files of package <1430312533>
PACKAGE => Checking free disk space for package <1430312533>
PACKAGE => Building ZIP for package <1430312533>
PACKAGE => Verifying ZIP signature for package <1430312533>
DOWNLOAD => Package <1430312533> built successfully
DOWNLOAD => Executing action <EXECUTE> for package <1430312533>
PACKAGE => Executing command <msiexec /x python.msi /qn> for package <1430312533>
PACKAGE => Package <1430312533> successfully executed. Command exit code is <0>
DOWNLOAD => Sending result code <SUCCESS> for package <1430312533>

```