

Machine	OS	Distribution	Version	C/S	IP
Poste 17	Linux	Debian8.5	3.0		192.168.1.136

HIRBEC  
Antoine

GESTION DE PARC INFORMATIQUE

BTS SIO 1<sup>ère</sup> A  
18/03/2016

# GESTION DE PARC INFORMATIQUE

Machine	OS	Distribution	Version	C/S	IP
Poste 17	Linux	Debian8.5	3.0		192.168.1.136

HIRBEC  
Antoine

GESTION DE PARC INFORMATIQUE

BTS SIO 1<sup>ère</sup> A  
18/03/2016

## Table des matières

Objectif(s) : .....	3
Introduction : .....	4
Création de la VM et continuité jusqu'à l'installation de OCSInventory. ....	5
Installation et configuration du service OCSInventory : .....	6
Console d'administration : .....	7
La collecte d'informations : .....	8
La collecte d'informations : .....	10
Configuration d'un serveur HTTPS : .....	13
Première étape : obtenir un certificat pour le serveur Web .....	14
Deuxième étape : configurer Apache2 avec mod_ssl : .....	15
Copier le certificat sur chaque client : .....	18
Le déploiement d'une application : .....	18

Machine	OS	Distribution	Version	C/S	IP
Poste 17	Linux	Debian8.5	3.0		192.168.1.136

HIRBEC  
Antoine

GESTION DE PARC INFORMATIQUE

BTS SIO 1<sup>ère</sup> A  
18/03/2016

## Objectif(s) :

L'objectif de cette procédure est mettre en place une gestion de parc informatique grâce à OCSInventory.

## Prérequis :

- VirtualBox
- Debian8.5
- Apache2
- OCSInventory

## Légende :

Les textes surlignés en jaune correspondent à des commandes ou à des indications qui permet de justifier les résultats obtenus ou de montrer des informations qu'elles doivent être respectées.

Machine	OS	Distribution	Version	C/S	IP
Poste 17	Linux	Debian8.5	3.0		192.168.1.136

HIRBEC  
Antoine

GESTION DE PARC INFORMATIQUE

BTS SIO 1<sup>ère</sup> A  
18/03/2016

## Introduction :

Le parc informatique d'une organisation est un assemblage, parfois hétéroclite de matériels et de logiciels accumulés tout au long des années.

On y trouve des :

- matériels différents (téléphones, portables, ordinateurs fixes, tablettes, imprimantes, éléments d'interconnexions) qui peuvent être de plusieurs générations
- logiciels et systèmes d'exploitation variés (Linux, Windows, Mac OS, etc.)
- applications utilisées dans différentes versions
- niveaux de sécurité hétérogènes

De plus, la quantité de matériels et de logiciels à gérer, leur éclatement au sein de l'organisation souvent très étendue dans l'espace, les exigences de performance et de réactivité font que la gestion de parc est devenue un processus global, complet et indispensable. Cette activité de gestion de parc informatique est décrite dans le processus ITIL1 Gestion des configurations.

Cette gestion de parc permet, d'une part, de répondre aux multiples questions quotidiennes posées à l'administrateur réseau (Quelles sont les versions de Windows installées et sur quels postes ? Y a-t-il des disques durs proches de la saturation ? Tel matériel est-il bien connecté au commutateur ? A quel endroit se trouve tel élément ? Quelle est la valeur actuelle de tel autre composant ? Quels sont les postes encore sous garantie ? ...

### **Le serveur OCS NG (Open Computer and Software Inventory Next Generation)**

Il 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) avec mise à jour automatique des éléments inventoriés
- de procéder à une gestion minimale du parc
- de téléistribuer des fichiers et des applications

### **Le serveur de gestion (Management server) comprend quatre composants principaux :**

- le serveur de base de données (Database server) qui stocke les informations d'inventaire.
- le serveur de communication (Communication server) qui gère les échanges entre les agents et le serveur de base de données.
- le serveur de déploiement (Deployment server) qui conserve les informations de configuration des paquets à télé-déployer.

Machine	OS	Distribution	Version	C/S	IP
Poste 17	Linux	Debian8.5	3.0		192.168.1.136

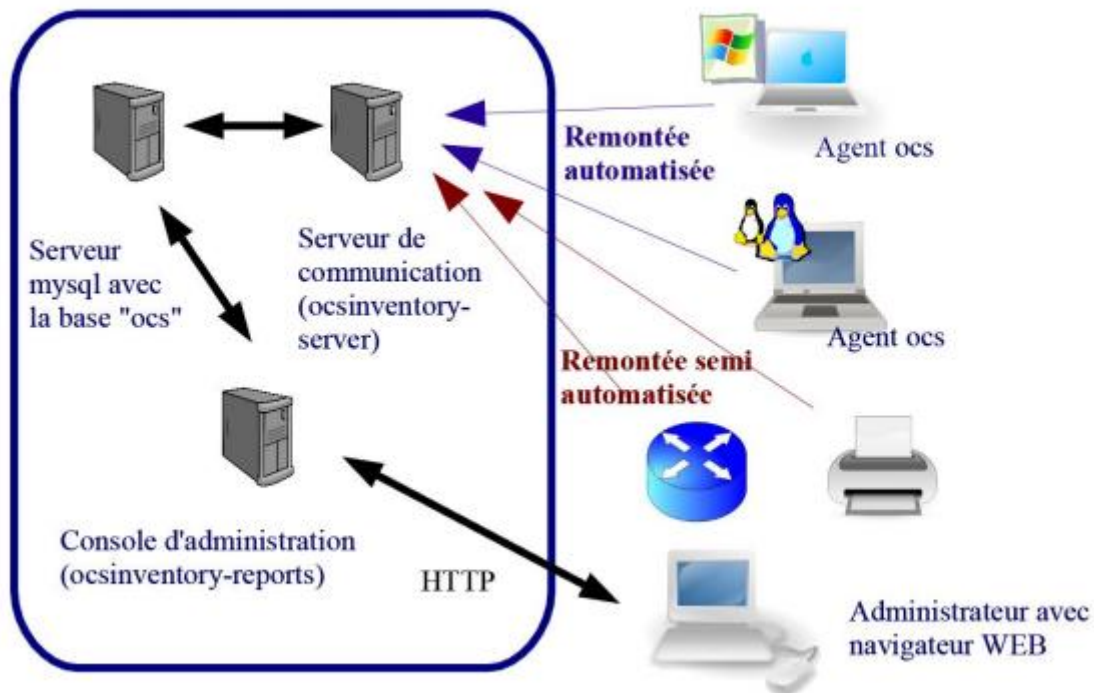
HIRBEC  
Antoine

GESTION DE PARC INFORMATIQUE

BTS SIO 1<sup>ère</sup> A  
18/03/2016

- la console d'administration (Administration console), accessible depuis une interface Web très intuitive, qui permet d'interroger la base de données.

**Schéma de collaboration des applications (sans le serveur de déploiement) :**



Création de la VM et continuité jusqu'à l'installation de OCSInventory.

On commence par mettre à jour les paquets : apt-get update.

Puis d'installer apache ; et mysql-server et mysql-client

Apt-get apache2

Apt-get mysql-server

Apt-get mysql-client.

Remettez à jour les paquets.

Machine	OS	Distribution	Version	C/S	IP
Poste 17	Linux	Debian8.5	3.0		192.168.1.136

HIRBEC  
Antoine

GESTION DE PARC INFORMATIQUE

BTS SIO 1<sup>ère</sup> A  
18/03/2016

## Installation et configuration du service OCSInventory :

Nous allons maintenant installer le service OCSInventory-server :

```
apt-get install ocsinventory-server ocsinventory-reports
```

Une fois les paquets installés, nous allons le configurer sur internet :

<http://@IPserveur/ocsreports>

On arrive donc ici :

OCS-NG Inventory Installation

DB configuration not completed. Automatic install launched

WARNING: You will not be able to build any deployment package with size greater than 100MB  
You must raise both `post_max_size` and `upload_max_filesize` in your `php.ini` to increase this limit.  
WARNING: If you change default database name (ocsweb), don't forget to update your ocs engine files (file `z-ocsinventory-server.conf`)

MySQL login:

MySQL password:

Name of Database:

MySQL HostName:

Information :

- **Les fichiers de configuration** de chacune des applications se trouvent dans `/etc/ocsinventory`
- Un répertoire "ocsinventory-server" est créé dans `/usr/share/`
- Un répertoire "ocsinventory-reports" est créé dans `/usr/share/` et dans `/var/lib/`
- La documentation de chacune des applications se trouve dans `/usr/share/doc/`
- La configuration pour le serveur WEB : `/etc/apache2/conf.d/ocsinventory.conf` et `/etc/apache2/conf.d/ocsreports.conf`
- Un utilisateur Mysql « ocs » qui a un certain nombre de droits sur cette base de données est créé par défaut avec comme mot de passe ocs.

Machine	OS	Distribution	Version	C/S	IP
Poste 17	Linux	Debian8.5	3.0		192.168.1.136

HIRBEC  
Antoine

GESTION DE PARC INFORMATIQUE

BTS SIO 1<sup>ère</sup> A  
18/03/2016

## Console d'administration :

La gestion du parc s'effectue via la console web d'administration. On accède à cette console avec l'URL suivante : <http://@IPserveur/ocsreports/> :

Par sécurité, il faut :

- supprimer ou renommer le fichier « install.php » qui est à la racine du serveur Web d'ocsreports
- modifier le mot de passe d'admin
- modifier le mot de passe de l'utilisateur mysql « ocs » et changer en conséquence dans les fichiers « dbconfig.inc.php » et « ocsinventory.conf » d'Apache2

Changer le mot de passe de admin

La page d'accueil de l'administration est la suivante :

The screenshot shows the OCS Inventory Reports administration interface. At the top left is the OCS logo. In the center, a red alert box reads: "ALERTE SECURITE! Le fichier install.php est présent dans votre répertoire d'interface. Le compte/mot de passe par défaut est actif sur votre base de donnée: ocsweb. Le compte/mot de passe par défaut de l'interface WEB est actif". To the right of the alert, there is a version indicator "Ver. 2.0" and a power icon. Below the header is a row of icons for navigation. A menu bar contains "ACTIVITE", "LOGICIEL", "MATERIEL", "DIVERS", and "MESSAGES". The main content area displays a summary table:

ACTIVITE	LOGICIEL	MATERIEL	DIVERS	MESSAGES
Machines en base	0			
Machines vues	0			
Machines ayant pris contact aujourd'hui	0			
Nombre d'inventaires aujourd'hui	0			
Machines absentes depuis plus de 30 jours	0			
Nombre de périphériques SNMP remontés	0			

Gestion des utilisateurs

On va donc supprimer et modifier les 3 choses dites juste au-dessus.

Pour le supprimer : `/usr/share/ocsinventory-reports`

- On va maintenant aller dans le module 'configuration' symbolisé avec une clé à molette.
  - Le but étant de ne pas trop charger le réseau, il faut éviter :
    - De faire des remontées constamment
    - De faire des remontées systématiques lors de chaque lancement du client ;
    - De faire les remontées de tous les clients en même temps.
  - Ce sont les paramètres PROLOG\_FREQ (onglet serveur) et FREQUENCY (onglet inventaire) qui gèrent le rythme des inventaires.
- On va mettre les logs dans le répertoire: `/var/log/ocsinventory-server/`, mais il faut au préalable les activer en positionnant à "on" la variable "LOGLEVEL".

Machine	OS	Distribution	Version	C/S	IP
Poste 17	Linux	Debian8.5	3.0		192.168.1.136

HIRBEC  
Antoine

GESTION DE PARC INFORMATIQUE

BTS SIO 1<sup>ère</sup> A  
18/03/2016

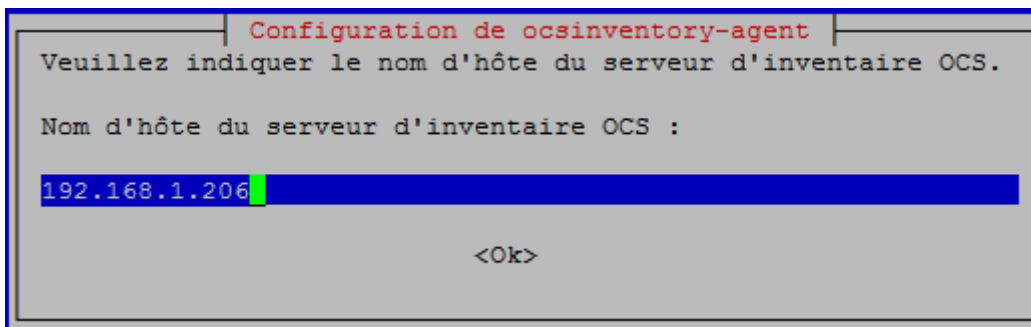
- **PROLOG\_FREQ** définit en nombre d'heure la **période max** entre 2 **lancements** d'un agent. Cette notion de "**période max**" permet d'éviter les surcharges si tous les postes remontaient leur inventaire simultanément.
  - C'est la valeur de la variable FREQUENCY qui va réellement permettre le lancement de l'inventaire :
    - **Toujours inventorié (always)** : la remontée sera réalisée sans condition dès que l'agent sollicite le serveur (c'est la valeur par défaut) ;
    - **Jamais inventorié (never)** : aucune remontée ne sera réalisée ;
    - **Personnalisé (custom)** : définit une **fréquence de remontée d'inventaire en nombre de jours** : la remontée sera réalisée lors de la sollicitation
      - **Ex** : FREQUENCY = toujours inventorié et PROLOG\_FREQ = 24 : toutes les 24 heures au maximum, on force une remontée qui sera faite systématiquement.

## La collecte d'informations :

Installation de l'agent sous Linux Debian :

```
#apt-get install ocsinventory-agent
```

Le système propose une configuration d'ocsinventory-agent. **Choisir la méthode "HTTP"** qui permet de remonter les informations à un serveur OCS,



### 3 fichiers sont créés :

- Un fichier de configuration "/etc/ocsinventory/ocsinventory-agent.cfg" dans lequel vous trouverez notamment le nom d'hôte (ou l'adresse IP) précisé précédemment.

Exemple de fichier ocsinventory-agent.cfg :

```
server=serveurDebian  
tag=Linux_Serveur
```



Machine	OS	Distribution	Version	C/S	IP
Poste 17	Linux	Debian8.5	3.0		192.168.1.136

HIRBEC  
Antoine

GESTION DE PARC INFORMATIQUE

BTS SIO 1<sup>ère</sup> A  
18/03/2016

Le "TAG" représente une rapide description de la machine (et permettra des recherches par catégorie) : s'il n'a pas été précisé lors de la configuration de l'agent, il peut être ajouté ou modifié via la console d'administration du serveur.

- Le fichier de rotation des logs : /etc/logrotate.d/ocsinventory-agent qui configure la rotation quotidienne des logs de l'agent OCS Inventory NG
- Un script pour l'agent (une tâche cron) : /etc/cron.daily/ocsinventory-agent ; ce script s'exécutera chaque jour à l'heure précisée dans /etc/crontab (0 heures 26 dans l'exemple ci-dessous) :  
26 0 \* \* \* root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )

La première remontée d'inventaire ne se fera qu'à l'heure indiquée et ensuite le rythme des remontées dépendra des valeurs des variables PROLOG\_FREQ et FREQUENCY définies préalablement.

**Pour forcer la remontée d'inventaire une première fois sans attendre le premier déclenchement du cron, il suffit d'exécuter la commande ocsinventory-agent.**

Cliquez ici, dans la console d'administration, pour voir l'ensemble des machines inventoriées

The screenshot shows the OCS Inventory NG administration interface. At the top, there are navigation icons. Below them, a search bar and a button labeled "5 Résultat(s) (Télécharger)". A table displays the following data:

Account info: TAG	△ Dernier inventaire	Machine	Utilisateur	Système	RAM(MB)	CPU(MHz)	Sélectionner	Supprimer
NA	2012-10-15 14:58:55	servWheezy	root	Debian GNU/Linux wheezy/sid	997	2321	<input type="checkbox"/>	✗
NA	2012-10-15 14:35:26	WIN7TEST32BITS	Administrateur	Microsoft Windows 7 Professionnel	2048	2394	<input type="checkbox"/>	✗
NA	2012-10-15 09:54:27	LLB-3DD2118A3B9	llb	Microsoft Windows XP Professional	1789	2000	<input type="checkbox"/>	✗
Linux_Serveur	2012-10-15 00:26:05	squeezeApoR1		Debian GNU/Linux 6.0.1	1024	2393	<input type="checkbox"/>	✗
NA	2012-10-09 15:47:37	BTSINFO-PC	Btsinfo	Microsoft Windows 7 Professionnel	1024	3000	<input type="checkbox"/>	✗

Après avoir tapé la commande permettant de déclencher la remontée d'inventaire on obtient cela :

The screenshot shows the OCS Inventory NG administration interface after a manual inventory scan. A button labeled "1 Résultat(s) (Télécharger)" is visible. A table displays the following data:

Account info: TAG	△ Dernier inventaire	Machine	Utilisateur	Système	RAM(MB)	CPU(MHz)	Sélectionner	Supprimer
NA	2016-05-04 15:59:21	HIRBEC-LETORT	root/antoine	Debian GNU/Linux 8.4 (jessie)	494	2333	<input type="checkbox"/>	✗

Un clic sur le nom d'une machine permet d'afficher, dans un autre onglet, les détails inventoriés du poste.

Machine	OS	Distribution	Version	C/S	IP
Poste 17	Linux	Debian8.5	3.0		192.168.1.136

HIRBEC  
Antoine

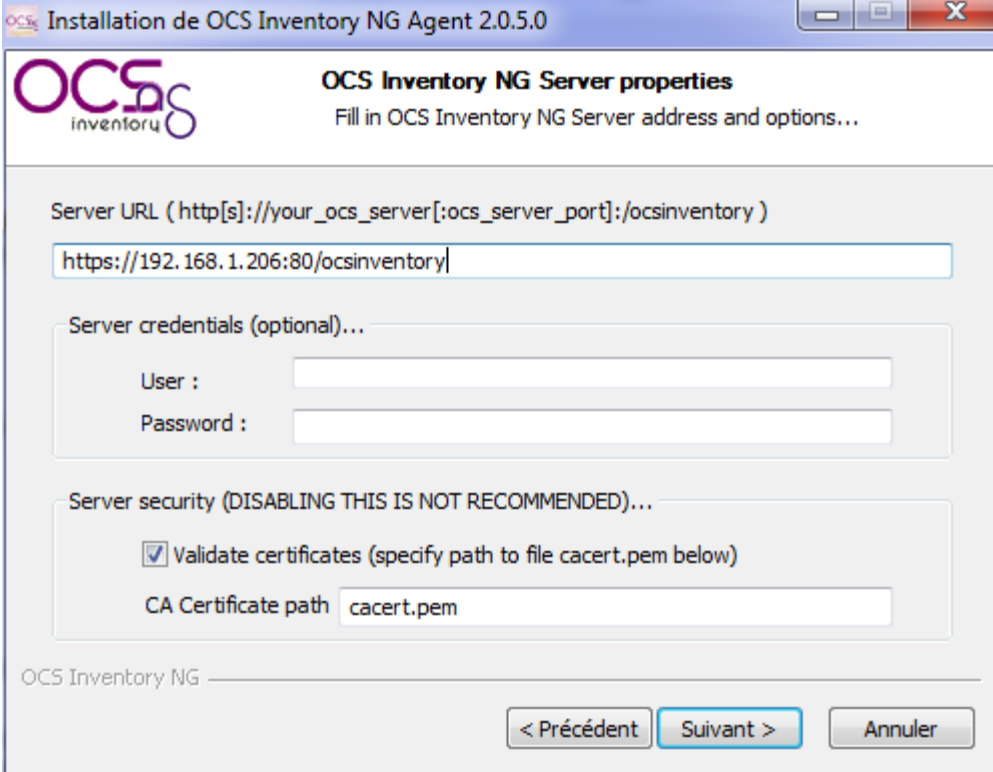
GESTION DE PARC INFORMATIQUE

BTS SIO 1<sup>ère</sup> A  
18/03/2016

## La collecte d'informations :

### Installation de l'agent sous Windows :

Après avoir lancé le .exe et validé la licence, on arrive sur cette fenêtre :



The screenshot shows a Windows installation window titled "Installation de OCS Inventory NG Agent 2.0.5.0". The window contains the OCS Inventory NG logo and the text "OCS Inventory NG Server properties" and "Fill in OCS Inventory NG Server address and options...".

The configuration fields are as follows:

- Server URL ( http[s]://your\_ocs\_server[:ocs\_server\_port]:/ocsinventory ) :
- Server credentials (optional)...
  - User :
  - Password :
- Server security (DISABLING THIS IS NOT RECOMMENDED)...
  - Validate certificates (specify path to file cacert.pem below)
  - CA Certificate path

At the bottom, there are three buttons: "< Précédent", "Suivant >" (highlighted in blue), and "Annuler".

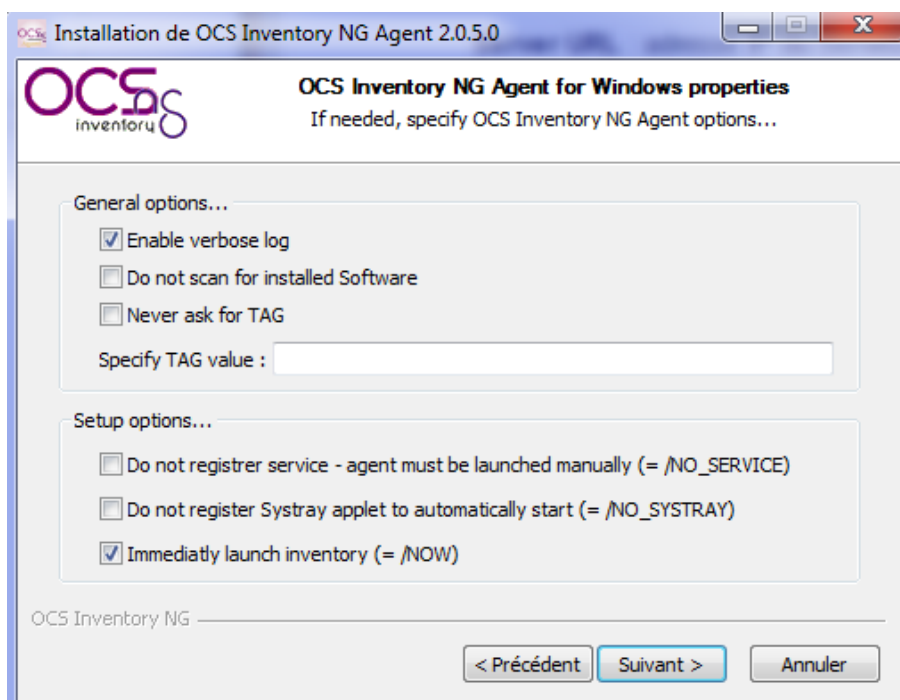
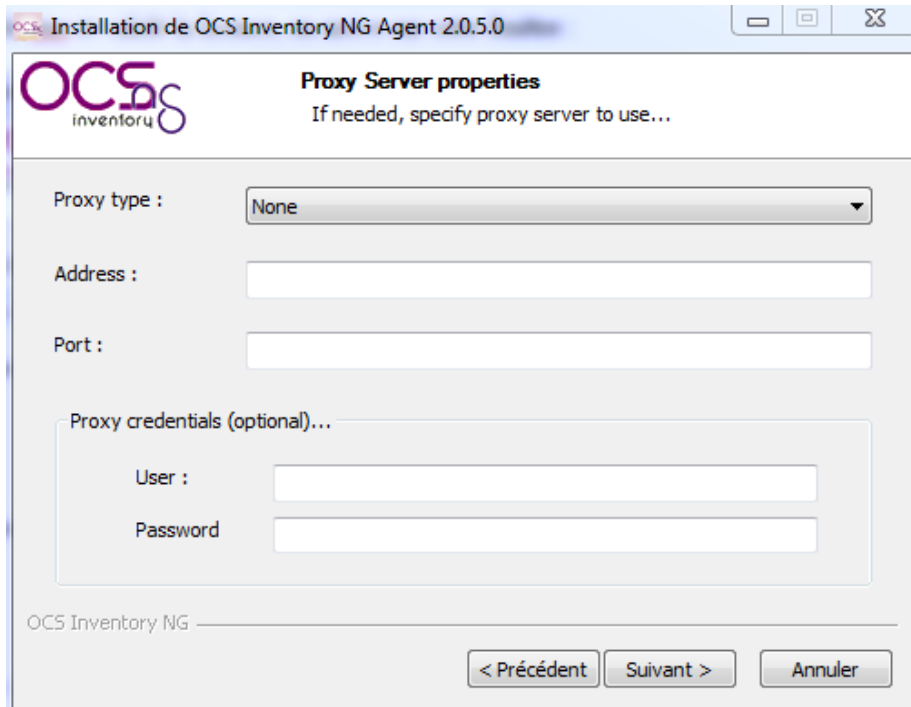
Machine	OS	Distribution	Version	C/S	IP
Poste 17	Linux	Debian8.5	3.0		192.168.1.136

HIRBEC  
Antoine

GESTION DE PARC INFORMATIQUE

BTS SIO 1<sup>ère</sup> A  
18/03/2016

Ensuite, on configure comme ceci :



Machine	OS	Distribution	Version	C/S	IP
Poste 17	Linux	Debian8.5	3.0		192.168.1.136

HIRBEC  
Antoine

GESTION DE PARC INFORMATIQUE

BTS SIO 1<sup>ère</sup> A  
18/03/2016

**Les répertoires d'installation** sont, par défaut

- C:\Program Files\OCS Inventory Agent\ pour les exécutables et dll
- C:\ProgramData\OCS Inventory NG\Agent\ pour les fichiers de configurations et les fichiers d'activité (log)

Une fois l'agent installé sur le client, **le service OCSinventory** est configuré pour être lancé automatiquement en tant que **service au démarrage**.

- Une fois l'inventaire lancé, il est possible de voir les logs :
  - C:\ProgramData\OCS Inventory NG\Agent\OCSInventory
- Le fichier ocsinventory lui permet de configurer l'agent

Machine	OS	Distribution	Version	C/S	IP
Poste 17	Linux	Debian8.5	3.0		192.168.1.136

HIRBEC  
Antoine

GESTION DE PARC INFORMATIQUE

BTS SIO 1<sup>ère</sup> A  
18/03/2016

```

1  [OCS Inventory Agent]
2  ComProvider=ComHTTP.dll
3  Debug=1
4  Local=
5  NoSoftware=0
6  HKCU=0
7  NoTAG=0
8  IpDisc=
9  [HTTP]
10 Server=http://192.168.1.206:80/ocsinventory
11 SSL=1
12 CaBundle=cacert.pem
13 AuthRequired=0
14 User=
15 Pwd=
16 ProxyType=0
17 Proxy=
18 ProxyPort=0
19 ProxyAuthRequired=0
20 ProxyUser=
21 ProxyPwd=
22 [OCS Inventory Service]
23 TTO_WAIT=86160
24 PROLOG_FREQ=24
25 OLD_PROLOG_FREQ=24
26

```

- Il y a aussi un fichier ocsinventory/log/bak :
  - Celui-ci est une sauvegarde des logs ( OCSInventory ).

## Configuration d'un serveur HTTPS :

Machine	OS	Distribution	Version	C/S	IP
Poste 17	Linux	Debian8.5	3.0		192.168.1.136

HIRBEC  
Antoine

GESTION DE PARC INFORMATIQUE

BTS SIO 1<sup>ère</sup> A  
18/03/2016

Le serveur HTTPS sera sur le même serveur physique que le serveur OCS

Les clés privée, publique et le certificat vont être créés avec l'utilitaire "OpenSSL" ; en conséquence, ce dernier doit être installé sur le serveur OCS.

## Première étape : obtenir un certificat pour le serveur Web

Il est possible :

- D'acheter un certificat SSL 128 bits à un prix raisonnable aux alentours de 50.00\$US/an.
- D'obtenir un certificat d'essai utilisable en environnement de test :
- De créer son certificat: c'est ce que nous nous proposons de faire... mais toute solution opérationnelle sera acceptée.

En prenant la 3eme option :

La documentation officielle d'OCS propose un script de génération de certificat à utiliser avec Apache reproduit ci-dessous qui fonctionne parfaitement :

```
GNU nano 2.2.6                                Fichier : apache_generate_cert.sh
#!/bin/sh
#
# En premier, generer le certificat requis
#
# Generer une clé RSA de 1024 bits, enregistrer la cle privée dans un
# fichier PEM de mot-de-passe non protege server.key, en utilisant
# le fichier de configuration par défaut d'openssl
#
echo
echo Generation de la clé privée du serveur Apache...
echo
openssl genrsa -out server.key 1024
#
# Maintenant, signez le certificat du serveur Apache avec
# la clé du serveur Apache
#
# Signez avec le certificat PEM server.crt,
# en utilisant le fichier PEM server.key pour clé privée du serveur,
# en utilisant le fichier de configuration par défaut d'openssl.
#
# Le certificat produit sera valide durant 1825 jours (soit 5 ans).
#
echo
echo Generation des certificats auto-signés du serveur Apache ...
echo
openssl req -outform PEM -new -key server.key -x509 -days 1825 -out server.crt
```

Ecrire le script ici :

```
root@HIRBEC-LETORT:~# nano apache_generate_cert.sh
```

Il faut maintenant :

- Mettre les droits d'exécution au script : `chmod u+x apache_generate_cert.sh`
- Puis lancer le script en utilisant cette commande : `sh apache_generate_cert.sh`

Machine	OS	Distribution	Version	C/S	IP
Poste 17	Linux	Debian8.5	3.0		192.168.1.136

HIRBEC  
Antoine

GESTION DE PARC INFORMATIQUE

BTS SIO 1<sup>ère</sup> A  
18/03/2016

Ce qui nous amène ici :

```

root@HIRBEC-LETORT:~# sh apache_generate_cert.sh

Generation de la cle privee du serveur Apache...

Generating RSA private key, 1024 bit long modulus
.....+++++
.....+++++
e is 65537 (0x10001)

Generation des certificats auto-signes du serveur Apache ...

You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:FR
State or Province Name (full name) [Some-State]:Normandie
Locality Name (eg, city) []:Caen
Organization Name (eg, company) [Internet Widgits Pty Ltd]:BTS_SIO_SISR
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:192.168.1.206
Email Address []:
root@HIRBEC-LETORT:~# █

```

## Deuxième étape : configurer Apache2 avec mod\_ssl :

- On va charger le **module SSL avec l'utilitaire Debian a2enmod** en lançant la commande `a2enmod ssl` dont le retour doit être du style : "Enabling module ssl. Run

Machine	OS	Distribution	Version	C/S	IP
Poste 17	Linux	Debian8.5	3.0		192.168.1.136

HIRBEC  
Antoine

GESTION DE PARC INFORMATIQUE

BTS SIO 1<sup>ère</sup> A  
18/03/2016

'/etc/init.d/apache2 restart' to activate new configuration!" ; ce que l'on pourra faire plus tard car il y a d'autres modifications à apporter.

L'activation du module a notamment pour effet d'activer le port d'écoute 443.

```
root@HIRBEC-LETORT:~# a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
service apache2 restart
```

cp

- On va copier le fichier du certificat du serveur « server.crt » et le fichier de la clé privée « server.key » dans les répertoires appropriés /etc/ssl/private/ :

#cp /root/server.\* /etc/ssl/private/

```
root@HIRBEC-LETORT:~# ls -l
total 12
-rwxr--r-- 1 root root 845 mai  9 14:28 apache_generate_cert.sh
-rw-r--r-- 1 root root 944 mai  9 14:38 server.crt
-rw-r--r-- 1 root root 891 mai  9 14:33 server.key
root@HIRBEC-LETORT:~# #cp /root/server.* /etc/ssl/private/
root@HIRBEC-LETORT:~# ls -l
total 12
-rwxr--r-- 1 root root 845 mai  9 14:28 apache_generate_cert.sh
-rw-r--r-- 1 root root 944 mai  9 14:38 server.crt
-rw-r--r-- 1 root root 891 mai  9 14:33 server.key
root@HIRBEC-LETORT:~# ls -l /etc/ssl/private/
total 4
-rw-r----- 1 root ssl-cert 1704 mai  2 17:11 ssl-cert-snakeoil.key
root@HIRBEC-LETORT:~# █
```

- On va ensuite mettre à jour les fichiers de configuration d'Apache2 pour utiliser ces fichiers (/etc/apache2/sites-available/default-ssl ou tout autre fichier de configuration personnel)



Machine	OS	Distribution	Version	C/S	IP
Poste 17	Linux	Debian8.5	3.0		192.168.1.136

HIRBEC  
Antoine

GESTION DE PARC INFORMATIQUE

BTS SIO 1<sup>ère</sup> A  
18/03/2016

Attention, par mesure de précaution faite une copie de ces fichiers avant de les modifier.

Les 2 directives à modifier sont SSLCertificateFile et SSLCertificateKeyFile :

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

Il faut donc aller dans nano /etc/apache2/sites-available/default-ssl.conf

Le sauvegarder ! Avec la commande cp

Et ensuite changer les deux directives !

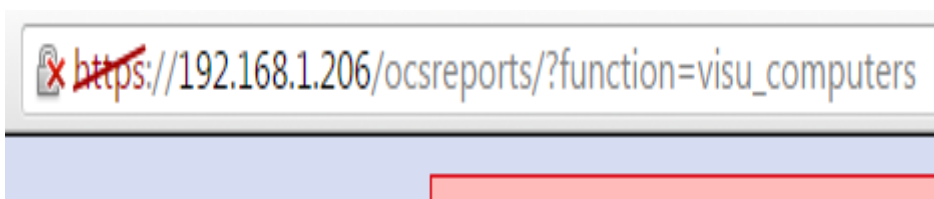
```
SSLCertificateFile /etc/ssl/private/server.crt
SSLCertificateKeyFile /etc/ssl/private/server.key
```

- On va maintenant activer la nouvelle configuration ! ++++++
- 1+0000000222
  - a2ensite default-ssl dont le retour doit être "Enabling site default-ssl.
  - Run '/etc/init.d/apache2 reload' to activate new configuration! "

```
root@HIRBEC-LETORT:/etc/apache2/sites-available# a2ensite default-ssl
Enabling site default-ssl.
To activate the new configuration, you need to run:
  service apache2 reload
root@HIRBEC-LETORT:/etc/apache2/sites-available# /etc/init.d/apache2 reload
[ ok ] Reloading apache2 configuration (via systemctl): apache2.service.
root@HIRBEC-LETORT:/etc/apache2/sites-available#
```

- Enfin pour finir on va redémarrer le service apache2 :
  - /etc/init.d/apache2 restart

```
root@HIRBEC-LETORT:/etc/apache2/sites-available# /etc/init.d/apache2 restart
[ ok ] Restarting apache2 (via systemctl): apache2.service.
root@HIRBEC-LETORT:/etc/apache2/sites-available#
```



L'URL :

<https://192.168.1.206/ocsreports/> devrait maintenant renvoyer une page d'alerte de sécurité ;

Machine	OS	Distribution	Version	C/S	IP
Poste 17	Linux	Debian8.5	3.0		192.168.1.136

HIRBEC  
Antoine

GESTION DE PARC INFORMATIQUE

BTS SIO 1<sup>ère</sup> A  
18/03/2016

ceci est normal car notre certificat n'a pas été signé par une autorité de certification connue. Il suffit d'accepter l'exception. Mais il est bien évident que ce n'est pas une solution acceptable en environnement de production.

## Copier le certificat sur chaque client :

**L'agent doit avoir un certificat pour valider l'authentification au serveur de déploiement.**

Il s'agit du fichier server.crt. Ce certificat doit être enregistré dans un fichier "cacert.pem" dans le répertoire de l'agent OCS Inventory NG sous Windows et dans le répertoire "/etc/ocsinventory-client" sous Linux.

```

-----BEGIN CERTIFICATE-----
MIICjDCCAfWgAwIBAgIJALnTe2d8gtnTMA0GCSqGSIb3DQEBCwUAMF8xCzAJBgNV
BAYTAKZSMRIwEAYDVQQIDAlOb3JtYW5kaWUxDTALBgNVBACMBENhZW4xFTATBgNV
BAoMDEJU19TSU9fU01TUjEWMBQGA1UEAwwNMTkyLjE2OC4xLjIwNjAeFw0xNjA1
MDkxMjM4NDhaFw0yMTA1MDgxMjM4NDhaMF8xCzAJBgNVBAYTAKZSMRIwEAYDVQQI
DA1Ob3JtYW5kaWUxDTALBgNVBACMBENhZW4xFTATBgNVBAoMDEJU19TSU9fU01T
UjEWMBQGA1UEAwwNMTkyLjE2OC4xLjIwNjCBnzANBkgqhkiG9w0BAQEFAAOBjQAw
gYkCgYEAy3oXCh93o9bTLo4yO3mQE3S2rca5i3kaxLHyiesSVkJtozDw/z1p+gOA
pJOH+LmjQmPttbcm7s/chQ7A5vBvXo/9LBnGYsXueGujAfp1f1nRAEZcbLZMzArI
m3tHhbthoBr2Ay07RnsPj81t65Nb7yvt8UJ3BiT6mg7C6q1QT+sCAwEAAANQME4w
HQYDVR0OBByEFNo2xPXj1xvX20ace7NsGnaZiprzMB8GA1UdIwQYMBaAFNo2xPXj
1xvX20ace7NsGnaZiprzMAwGA1UdEwQFMAMBAf8wDQYJKoZIhvcNAQELBQADgYEA
q1L+vbnZiOEAR7fx32I8G12WbP+1BSI5ifTX01e08T9Abbw51whvBVg+wSCynj7Z
8u2UAenJlomAw5ix0O6b2UTW0FuEsYs3FOc+h1nPzPdndEm1/SWD1grLhpzOFU/9
uAIGIG8IJFtO1YFkCJZFA5iyybf3pHwgrw/10/sG504=
-----END CERTIFICATE-----

```

Sur la machine linux, on est serveur et non client, pas besoin de copier le certificat.

## Le déploiement d'une application :

L'installation, la mise à jour et la suppression d'applications font partie du travail quotidien d'un administrateur réseau. Lorsque le parc de machine s'agrandit, **il devient très intéressant d'automatiser cette tâche.**

Préalables à exécuter :

Machine	OS	Distribution	Version	C/S	IP
Poste 17	Linux	Debian8.5	3.0		192.168.1.136

HIRBEC  
Antoine

GESTION DE PARC INFORMATIQUE

BTS SIO 1<sup>ère</sup> A  
18/03/2016

- Sur le serveur OCS, le déploiement d'application n'est pas activé par défaut; IL faut mettre le paramètre DOWNLOAD à "on" (menu "Configuration", onglet "Teledeploiement"); mettre aussi 15 secondes à DOWNLOAD\_PERIOD\_LATENCY (temps d'attente entre 2 périodes de télé-déploiement).



Inventory	Server	IpDiscover	Deployment	Redistribution Servers	Groups	Registry
Inventory files	Filters	Webservice	Interface	LDAP configuration	Snmp	Support
<b>DOWNLOAD</b> <i>Automatic software distribution functionality</i>		<input checked="" type="radio"/> ON <input type="radio"/> OFF				
<b>DOWNLOAD_CYCLE_LATENCY</b> <i>Waiting time between 2 cycles of deployment</i>		<input type="text" value="60"/> seconds <i>(Must be greater than or equal to 1)</i>				
<b>DOWNLOAD_FRAG_LATENCY</b> <i>Waiting time between 2 downloaded fragments</i>		<input type="text" value="10"/> seconds <i>(Must be greater than or equal to 1)</i>				
<b>DOWNLOAD_GROUPS_TRACE_EVENTS</b> <i>Specify if you want to track packages affected to a group on computer's level</i>		<input checked="" type="radio"/> ON <input type="radio"/> OFF				
<b>DOWNLOAD_PERIOD_LATENCY</b> <i>Waiting times between 2 periods of deployment</i>		<input type="text" value="15"/> seconds <i>(Must be greater than or equal to 1)</i>				

- Il faut aussi modifier le chemin du répertoire "download" qui est le répertoire par défaut où les agents vont télécharger le paquet ou les fragments de paquet ainsi qu'un fichier XML d'information : dans les variables DOWNLOAD\_URI\_FRAG et DOWNLOAD\_URI\_INFO, changer localhost par l'adresse IP du serveur.

<b>DOWNLOAD_URI_FRAG</b> <i>Address to find teledeploy packages fragments to activate</i>	<input type="radio"/> By default (HTTP://localhost/download) <input checked="" type="radio"/> Customize http:// <input type="text" value="192.168.1.206/download"/>
<b>DOWNLOAD_URI_INFO</b> <i>Address to find teledeploy packages INFO files to activate</i>	<input type="radio"/> By default (HTTPS://localhost/download) <input checked="" type="radio"/> Customize https:// <input type="text" value="192.168.1.206/download"/>

Machine	OS	Distribution	Version	C/S	IP
Poste 17	Linux	Debian8.5	3.0		192.168.1.136

HIRBEC  
Antoine

GESTION DE PARC INFORMATIQUE

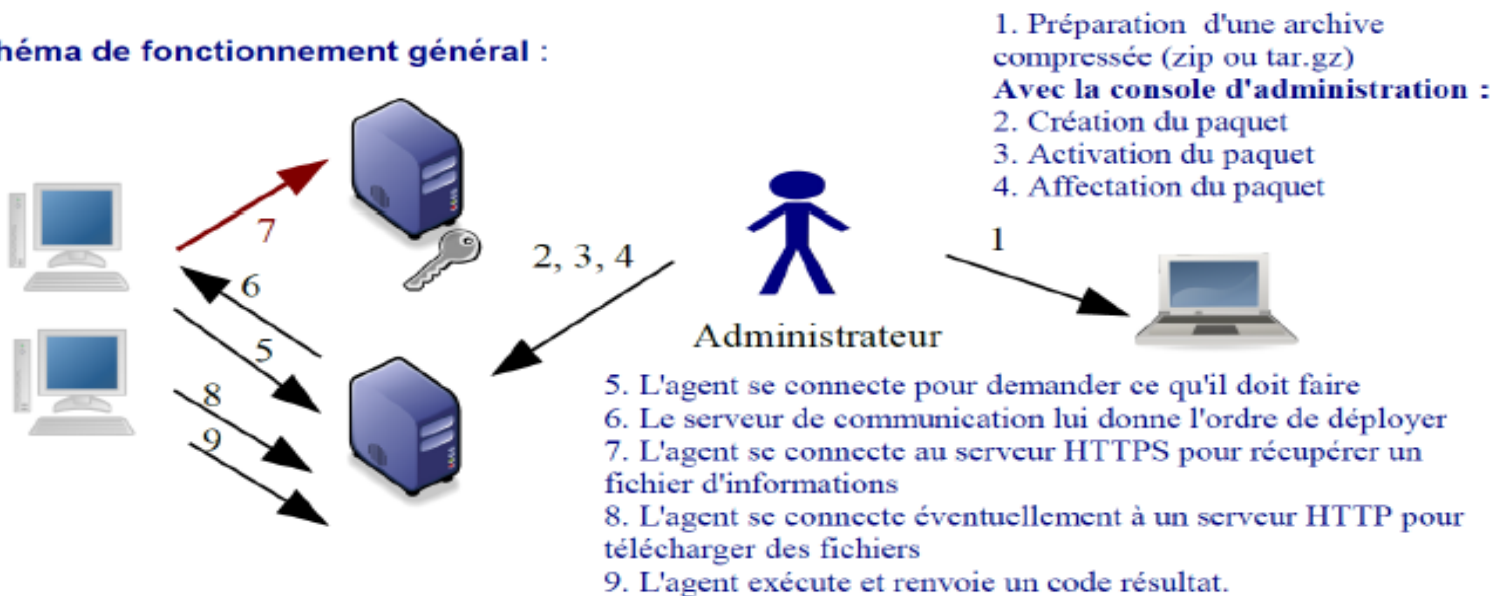
BTS SIO 1<sup>ère</sup> A  
18/03/2016

• Le déploiement d'applications ne fonctionne qu'avec un serveur Web sécurisé utilisant le protocole HTTPS basé sur l'authentification SSL, il est donc nécessaire de configurer le serveur HTTPS basé sur l'authentification SSL (annexe 5).

➤ Fait à chapitre d'avant !

Schéma de fonctionnement général :

**Schéma de fonctionnement général :**



Le principe de base est le suivant :

Comme nous l'avons déjà vu, l'agent se connecte au serveur de communication par le protocole HTTP pour lui demander ce qu'il doit faire. En fonction de sa configuration, le serveur peut répondre :

- d'envoyer un inventaire ;
- de découvrir le réseau avec le service IpDiscovery ;
- de déployer un ou plusieurs paquets ;
- de ne rien faire.

➤ Lorsque l'agent a l'ordre de **déployer un paquet**, il contacte **via le protocole HTTPS** le serveur de déploiement **afin d'y récupérer un fichier d'informations** (IDA : "Instruction Déploiement d'Applications") associé qui est un **fichier XML décrivant le paquet et l'action que l'agent devra exécuter**. C'est un fichier qui dispose d'un champ d'action important d'où la nécessité de sécuriser et d'authentifier le serveur sur lequel il se trouve.

Machine	OS	Distribution	Version	C/S	IP
Poste 17	Linux	Debian8.5	3.0		192.168.1.136

HIRBEC  
Antoine

GESTION DE PARC INFORMATIQUE

BTS SIO 1<sup>ère</sup> A  
18/03/2016

- L'agent devra éventuellement télécharger, via le protocole HTTP, **un fichier ou des fragments de fichiers** (ce dernier point est optionnel si les instructions ne consistent qu'à exécuter une ou plusieurs commandes).

### **L'administrateur devra au préalable :**

- préparer une archive compressée (en .ZIP pour Windows et en .tar.gz pour Linux) des fichiers nécessaires ;
  - On va créer un fichier test, l'enregistrer sous et lui donner comme nom, test.zip ainsi le fichier sera une archive compressé.

- créer le paquet grâce à la console d'administration ;

- **Le nom du paquet à déployer** auquel sera associé un identifiant unique dans la base de données.

- **Le système** : il est possible de déployer des paquets sur Windows ou sur Linux.

- **Le protocole** utilisé pour le transfert des données est HTTP. La gestion de parc informatique Page 15 **la priorité** permet de définir quels paquets doivent être installés avant d'autres. Au total 11 niveaux de priorité sont disponibles. Plus le chiffre défini comme priorité est bas, plus la priorité sera forte.

La priorité 0 est donc la plus forte, mais attention, celle-ci doit être utilisée avec précaution car un paquet ayant cette priorité devra obligatoirement se déployer correctement sinon l'ensemble des autres paquets ne sera pas déployé. La priorité 5 (proposée par défaut) convient la plupart du temps.

**Le fichier** : selon la documentation, tous les paquets doivent être compressés en ZIP pour l'agent Windows et en TAR.GZ pour les ordinateurs Linux. **Cette archive compressée doit donc être préparée préalablement.**

- Trois types d'**actions** sont disponibles:
  - **Lancer** : pour déployer un fichier ZIP ou TAR.GZ et lancer avec ou sans paramètre un fichier exécutable **incluant** un fichier ZIP ou TAR.GZ. Le fichier ZIP ou TAR.GZ sera décompressé dans un répertoire temporaire, et la commande associée (le nom du fichier exécutable sans le chemin !) sera lancée dans le répertoire temporaire.
  - **Exécuter** : pour déployer un fichier ZIP ou TAR.GZ (optionnellement), et lancer avec ou sans paramètre un fichier exécutable **incluant ou non** un fichier ZIP ou TAR.GZ. Si l'exécutable n'est pas inclus dans le fichier ZIP ou TAR.GZ, il doit être une partie de logiciel toujours installée dans l'ordinateur client. Typiquement, cela peut être une commande Windows standard tel qu'un appel de l'installateur Windows, commande RPM, DPKG ou TAR.GZ sur Linux. Le fichier ZIP ou TAR.GZ sera décompressé dans un répertoire temporaire, et la commande associée (le nom du fichier exécutable avec le chemin ou les paramètres si besoin) sera lancée dans le répertoire temporaire.

Machine	OS	Distribution	Version	C/S	IP
Poste 17	Linux	Debian8.5	3.0		192.168.1.136

HIRBEC  
Antoine

GESTION DE PARC INFORMATIQUE

BTS SIO 1<sup>ère</sup> A  
18/03/2016

- **Stocker** : pour déployer un fichier ZIP ou TAR.GZ et enregistrer seulement son contenu dans un enregistrement de son ordinateur client. Il faut donc donner le chemin de stockage sur l'ordinateur client.

### Fabrication d'un paquet de télédéploiement

**Création d'un nouveau paquet**

Nom:	Utilitaire_putty
Description:	Client ssh putty
Système:	WINDOWS ▾
Protocole:	HTTP ▾
Priorité:	5 ▾
Fichier (déployé sur les ordinateurs clients):	C:\Documents and Settir <span style="float: right;">Parcourir..</span>
Action:	<span>Stocker ▾</span> Chemin: <input style="width: 150px;" type="text" value="C:\Program Files\"/>

**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 ▾

Machine	OS	Distribution	Version	C/S	IP
Poste 17	Linux	Debian8.5	3.0		192.168.1.136

HIRBEC  
Antoine

GESTION DE PARC INFORMATIQUE

BTS SIO 1<sup>ère</sup> A  
18/03/2016

New package building [utilitaire_putty]	
File name:	test.zip
Unique identifier:	1462803717
Digest MD5 / Hexa:	cbe45e992dc84b765a370f9c7e1d0a67
Total size:	0 KB
Fragments size (1 KB min):	1 <input type="text"/> KB
Fragments number:	1 <input type="text"/>
Estimated time for deploy:	<input type="text"/>

Send

➤ activer le paquet ;



Package activation

Packages created manually

Show: 20

Restrict view:  Filter

Add column:

1 Result(s) (Download)

Timestamp	Creation date	Name	Notified	Success	Error	Archives	Activate	Delete	Select	Stats
1462803717	2016-05-09 16:21:57	utilitaire_putty							<input type="checkbox"/>	



Or activate a package manually Timestamp:

On arrive sur cette petit fenetre :

Machine	OS	Distribution	Version	C/S	IP
Poste 17	Linux	Debian8.5	3.0		192.168.1.136

HIRBEC  
Antoine

GESTION DE PARC INFORMATIQUE

BTS SIO 1<sup>ère</sup> A  
18/03/2016

The screenshot shows the OCS Inventory web interface in Google Chrome. The browser address bar displays the URL: [https://192.168.1.206/ocsreports/index.php?function=tele\\_popup\\_active&head=1&active=1462803717](https://192.168.1.206/ocsreports/index.php?function=tele_popup_active&head=1&active=1462803717). The page content includes a dropdown menu labeled "Activate:" with "MANUAL" selected. Below this, a blue box contains the text "Package activation => utilitaire\_putty (1462803717)". Underneath, there are two input fields: "Fragments url" with the value "http://192.168.1.206/download/1462803717" and "Https url" with the value "https://192.168.1.206/download/1462803717". At the bottom of the blue box are two circular icons: a green checkmark and a red 'X'. Below the blue box, a white dialog box is open with the text "192.168.1.206 indique : Package activated, it can now be affected" and an "OK" button.

- Affecter le paquet aux machines sur lesquelles le déploiement doit s'effectuer
  - Une fois le paquet activé, il est possible de le déployer facilement sur un nombre important de postes.
  - Le plus simple est d'effectuer une recherche (dans notre exemple les ordinateurs qui exécutent un système d'exploitation Windows) ou utiliser un groupe dynamique ou statique puis de cliquer sur le lien l'icône "**Teledeployer**".

On va afficher toute les machines qui sont sur Windows.



Machine	OS	Distribution	Version	C/S	IP
Poste 17	Linux	Debian8.5	3.0		192.168.1.136

HIRBEC  
Antoine

GESTION DE PARC INFORMATIQUE

BTS SIO 1<sup>ère</sup> A  
18/03/2016

On recherche grace au filtre operating system : windows.



Show:    
 Add column:

**1 Result(s)** (Download)

Account info: TAG	Machine(s): Last contact	Machine(s): Last inventory	Machine(s): Operating system	Machine(s): User	Machine(s): User agent	Computer	Delete	Select
Salle 105	2016-05-09 14:03:59	2016-05-09 14:03:59	Microsoft Windows 7 Professionnel	etudiantsio	OCS-NG_WINDOWS_AGENT_v2.0.5.0	POSTE21	<input type="checkbox"/>	<input type="checkbox"/>



Choose a parameter: ... Choose ...   
 computers: Operating system

Cliquez sur "Select" et les postes clients seront avertis (notifiés) qu'un paquet doit être déployé dès leur prochaine communication avec le serveur (que l'on peut/doit provoquer...).

Ils téléchargeront alors dans un premier temps le fichier d'instructions XML et ensuite, s'il y a lieu, les différents fragments du paquet.

**Action on the query result**

**Packages on computers** | **Packages on servers groups**

Show:    
 Restrict view:     
 Add column:

**1 Result(s)** (Download)

Package Name	Priority	OS Name	SIZE (KB)	PACK_LOC	SELECT
utilitaire_putty	5	WINDOWS	56	192.168.1.206/download	<input type="checkbox"/>

Machine	OS	Distribution	Version	C/S	IP
Poste 17	Linux	Debian8.5	3.0		192.168.1.136

HIRBEC  
Antoine

GESTION DE PARC INFORMATIQUE

BTS SIO 1<sup>ère</sup> A  
18/03/2016

i
Action on the query result

✓
1 computer(s) successfully affected(s)

Packages on computers
Packages on servers groups

Show:

Restrict view:

Filter ✗

Add column:

**1 Result(s) (Download)**

Package Name	Priority <span style="color: red;">✗</span>	OS Name <span style="color: red;">✗</span>	SIZE (KB) <span style="color: red;">✗</span>	PACK_LOC	SELECT
utilitaire_putty	5	WINDOWS	56	192.168.1.206/download	⏪

On va maintenant refaire un inventaire et regarder dans els logs what's happens ?

```

Download - Bloc-notes
Fichier  Edition  Format  Affichage  ?
=====
Starting OCS Inventory NG Package Download and Setup Tool on Monday, May 09, 2016 16:50:21.
DOWNLOAD => Running OCS Inventory NG Download Version 2.0.5.0
DOWNLOAD => Using OCS Inventory NG Framework Version 2.0.5.0
DOWNLOAD => Using network connection with communication server
COM PROVIDER => Loading Communication Provider <C:\Program Files (x86)\OCS Inventory Agent\ComHTTP.dll>
DOWNLOAD => Using Communication Provider <OCS Inventory NG CURL Communication Provider> Version <2.0.5.0>
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: 00:00:00.

```

Machine	OS	Distribution	Version	C/S	IP
Poste 17	Linux	Debian8.5	3.0		192.168.1.136

HIRBEC  
Antoine

GESTION DE PARC INFORMATIQUE

BTS SIO 1<sup>ère</sup> A  
18/03/2016

Dans les logs on voit cela :

```

INVENTORY => Writing new inventory state
AGENT => Communication Server ask for Package Download
DOWNLOAD => Package history file successfully cleaned for duplicate IDs
DOWNLOAD => Metadata file <info> for package <1462803717> is located at <https://192.168.1.206/download/1462803717/info>
COM SERVER => Initializing CURL library for getFile
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 fileGet request to URL <https://192.168.1.206/download/1462803717/info>
COM SERVER => fileGet response received <HTTP Status Code #403>
COM SERVER => Cleaning CURL library
ERROR *** DOWNLOAD => Failed to download Metadata file <https://192.168.1.206/download/1462803717/info> to <C:\Program Files\OCSInventory\ocsinventory-agent\bin>
DOWNLOAD => Download and setup tool successfully started
AGENT => unloading communication provider
AGENT => unloading plug-in(s)
AGENT => Execution duration: 00:00:37.

```

Le client ne peut pas récupérer le paquet

- Il faut aller dans les configurations d'apache2.
- Et mettre « allow from all » et mettre en commentaire « Require all denied ».
- Penser à restart le service apache2
- Et recommencer à affecter le paquet.

```

<Directory />
    Options FollowSymLinks
    AllowOverride None
    Allow from all
    #Require all denied
</Directory>

```

Photos des logs :

OCSInventory :

```

-----
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://192.168.1.206:80/ocsinventory>
COM SERVER => HTTP Post response received <HTTP Status Code #200>
COM SERVER => Cleaning CURL library
AGENT => Inventory successfully sent
INVENTORY => Writing new inventory state
AGENT => Communication Server ask for Package Download
DOWNLOAD => Package history file successfully cleaned for duplicate IDs
DOWNLOAD => Metadata file <info> for package <1462803717> is located at <https://192.168.1.206/download/1462803717/info>
COM SERVER => Initializing CURL library for getFile
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 fileGet request to URL <https://192.168.1.206/download/1462803717/info>
COM SERVER => fileGet response received <HTTP Status Code #200>
COM SERVER => Cleaning CURL library
DOWNLOAD => unloading communication provider
DOWNLOAD => Retrieve info file...OK (pack 1462803717)
DOWNLOAD => Package <1462803717> added to download queue
DOWNLOAD => Download and setup tool successfully started
AGENT => unloading communication provider
AGENT => unloading plug-in(s)
AGENT => Execution duration: 00:00:39.

```

Machine	OS	Distribution	Version	C/S	IP
Poste 17	Linux	Debian8.5	3.0		192.168.1.136

HIRBEC  
Antoine

GESTION DE PARC INFORMATIQUE

BTS SIO 1<sup>ère</sup> A  
18/03/2016

Download :

```

=====
Starting OCS Inventory NG Package Download and Setup Tool on Monday, May 09, 2016 17:16:17.
DOWNLOAD => Running OCS Inventory NG Download Version 2.0.5.0
DOWNLOAD => Using OCS Inventory NG Framework Version 2.0.5.0
DOWNLOAD => Using network connection with Communication Server
COM PROVIDER => Loading Communication Provider <C:\Program Files (x86)\OCS Inventory Agent\ComHTTP.dll>
DOWNLOAD => Using Communication Provider <OCS Inventory NG CURL Communication Provider> Version <2.0.5.0>
DOWNLOAD => Starting new period of 10 cycles
DOWNLOAD => Flushing package queue
DOWNLOAD => Parsing directory <C:\ProgramData\OCS Inventory NG\Agent\download> for packages
DOWNLOAD => Verifying package <C:\ProgramData\OCS Inventory NG\Agent\download\1462803717>
DOWNLOAD => Package <1462803717> verified and added to process queue
DOWNLOAD => Processing packages for cycle 1 on Monday, May 09, 2016 17:16:17
DOWNLOAD => Pausing for cycle latency (60 seconds)
DOWNLOAD => Processing packages for cycle 2 on Monday, May 09, 2016 17:17:17
DOWNLOAD => Pausing for cycle latency (60 seconds)
DOWNLOAD => Processing packages for cycle 3 on Monday, May 09, 2016 17:18:17
DOWNLOAD => Pausing for cycle latency (60 seconds)
DOWNLOAD => Processing packages for cycle 4 on Monday, May 09, 2016 17:19:17
DOWNLOAD => Pausing for cycle latency (60 seconds)
DOWNLOAD => Processing packages for cycle 5 on Monday, May 09, 2016 17:20:17
DOWNLOAD => Downloading package fragment <1462803717-1>
COM SERVER => Initializing CURL library for getFile
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 fileget request to URL <HTTP://192.168.1.206/download/1462803717/1462803717-1>
COM SERVER => fileget response received <HTTP Status Code #200>
COM SERVER => Cleaning CURL library
DOWNLOAD => Pausing for fragment latency (10 seconds)
DOWNLOAD => Pausing for cycle latency (60 seconds)
DOWNLOAD => Processing packages for cycle 6 on Monday, May 09, 2016 17:21:28
DOWNLOAD => Pausing for cycle latency (60 seconds)
DOWNLOAD => Processing packages for cycle 7 on Monday, May 09, 2016 17:22:28
DOWNLOAD => Pausing for cycle latency (60 seconds)
DOWNLOAD => Processing packages for cycle 8 on Monday, May 09, 2016 17:23:28
DOWNLOAD => Pausing for cycle latency (60 seconds)
DOWNLOAD => Processing packages for cycle 9 on Monday, May 09, 2016 17:24:28
DOWNLOAD => Pausing for cycle latency (60 seconds)
DOWNLOAD => Processing packages for cycle 10 on Monday, May 09, 2016 17:25:28
DOWNLOAD => Building package <1462803717>
PACKAGE => Verifying fragment files of package <1462803717>
PACKAGE => Checking free disk space for package <1462803717>
PACKAGE => Building ZIP for package <1462803717>
PACKAGE => Verifying ZIP signature for package <1462803717>
DOWNLOAD => Package <1462803717> built successfully
DOWNLOAD => Executing action <STORE> for package <1462803717>

```

Tout s'est passé correctement !